

工业和信息化部职业教育教学指导委员会“十二五”规划教材
全国高等职业教育计算机系列规划教材

Visual FoxPro 项目案例教程

（第2版）

丛书编委会

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书针对高职高专学生的特点,按照任务驱动和项目教学的思路进行编写,通过创建学生较为熟悉的“学生学籍管理系统”项目,将 Visual FoxPro 9.0 的相关知识,由浅入深地融入到 9 个项目当中,让学生在“会学”的基础上真正实现“会用”。同时,兼顾计算机等级考试,本书涵盖了全国计算机等级考试大纲的全部知识点。本书前 8 个项目按照基础知识、实例训练、项目分析、模仿设计与自主设计、项目小结、项目习题的顺序依次展开,分别介绍了 VFP 语言基础,数据库基础,数据表的建立及操作,查询、视图及 SQL 语言,程序设计和程序结构,表单,菜单,报表创建及修改等知识;最后一个项目结合前面所有知识,通过“公司人事管理系统”的实例,介绍了完整的应用系统开发流程。

本书适合作为高职高专院校相关专业的 VFP 课程教材和等级考试教材,也可作为具有计算机数据库基础知识的读者、计算机程序设计人员及计算机爱好者的参考书。本书配有电子课件、习题答案、源代码等资源,有需要的读者可以登录华信教育资源网(www.hxedu.com.cn)免费下载。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Visual FoxPro 项目案例教程/《全国高等职业教育计算机系列规划教材》丛书编委会编. —2 版. —北京:电子工业出版社,2011.9

(工业和信息化部职业教育指导委员会“十二五”规划教材 全国高等职业教育计算机系列规划教材)

ISBN 978-7-121-14511-7

I. ①V… II. ①全… III. ①关系数据库—数据库管理系统, Visual FoxPro—程序设计—高等职业教育—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2011)第 178607 号

策划编辑:左 雅

责任编辑:左 雅

印 刷:北京市李史山胶印厂

装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1 092 1/16 印张:16.5 字数:422.4 千字

印 次:2011 年 9 月第 1 次印刷

印 数:4000 册 定价:29.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

丛 书 编 委 会

主 任 郝黎明 逢积仁

副 主 任 左 雅 方一新 崔 炜 姜广坤 范海波 敖广武 徐云晴 李华勇

委 员 (按拼音排序)

陈国浪	迟俊鸿	崔爱国	丁 倩	杜文洁	范海绍	何福男
贺 宏	槐彩昌	黄金栋	蒋卫祥	李 琦	刘宝莲	刘红军
刘 凯	刘兴顺	刘 颖	卢锡良	孟宪伟	庞英智	钱 哨
乔国荣	曲伟峰	桑世庆	宋玲玲	王宏宇	王 华	王晶晶
温丹丽	吴学会	邢彩霞	徐其江	严春风	姚 嵩	殷广丽
尹 辉	俞海英	张洪明	张 薇	赵建伟	赵俊平	郑 伟
周绯非	周连兵	周瑞华	朱香卫	邹 羚		

本 书 编 委 会

主 编 梁玉凤 赵 娅

副 主 编 党进才 蔡 红

丛书编委会院校名单

(按拼音排序)

保定职业技术学院	山东省潍坊商业学校
渤海大学	山东司法警官职业学院
常州信息职业技术学院	山东信息职业技术学院
大连工业大学职业技术学院	沈阳师范大学职业技术学院
大连水产学院职业技术学院	石家庄信息工程职业学院
东营职业学院	石家庄职业技术学院
河北建材职业技术学院	苏州工业职业技术学院
河北科技师范学院数学与信息技术学院	苏州托普信息职业技术学院
河南省信息管理学校	天津轻工职业技术学院
黑龙江工商职业技术学院	天津市河东区职工大学
吉林省经济管理干部学院	天津天狮学院
嘉兴职业技术学院	天津铁道职业技术学院
交通运输部管理干部学院	潍坊职业学院
辽宁科技大学高等职业技术学院	温州职业技术学院
辽宁科技学院	无锡旅游商贸高等职业技术学校
南京铁道职业技术学院苏州校区	浙江工商职业技术学院
山东滨州职业学院	浙江同济科技职业学院
山东经贸职业学院	

前 言

Visual FoxPro 9.0 是 Microsoft 公司推出的 Visual FoxPro 的最新版本，是一个优秀的可视化数据编程工具，主要用于小型数据库系统的开发。无论是组织信息、运行查询、创建集成的关系型数据库系统，还是为最终用户编写功能全面的数据管理应用程序，Visual FoxPro 9.0 都可以提供管理数据所需的工具，可以在应用程序或 Visual FoxPro 数据库开发的任何一个领域中提供帮助。

本书作为高职高专的教学用书，是根据当前高职高专学生和教学环境的现状，结合职业需求，采用“工学结合”的思路编写的，并以学生较为熟悉的典型案例“学生学籍管理系统”贯穿全书。本书打破了传统的章节式编排，共分为 9 个项目，前 8 个项目按照基础知识、实例训练、项目分析、模仿设计与自主设计、项目小结、项目习题的顺序依次展开，分别介绍了 VFP 语言基础，数据库基础，数据表的建立及操作，查询、视图及 SQL 语言，程序设计和程序结构，表单，菜单，报表创建及修改等知识；最后一个项目是综合项目，通过一个“公司人事管理系统”的实例，介绍了完整的应用系统开发流程。针对社会上的等级考试，本书还在最后附上了 2010 年 9 月计算机二级 VFP 笔试试题、参考答案，2011 年计算机二级 VFP 考试大纲。通过本书的学习，学生可以将知识融会贯通于操作，在“会学”的基础上真正实现“会用”，提高利用 Visual FoxPro 9.0 解决实际问题的能力。

在每个项目中，任务 1 列出相关知识供学生查阅，讲解该项目涉及的知识点；任务 2 “实例训练”将项目中的主要功能拆分为能够独立运行的训练模块，并在训练中提出设计要求和预期目标等；任务 3 在“项目分析”中阐明模块的设计思想及技术要点，指出创新思路，给出设计提示与实现步骤，指明知识拓展方向，使学生了解完整项目的开发流程；任务 4 “模仿设计与自主设计”提供学生模仿设计与自主设计的习题，自主设计可参照书中项目进行。本书尽量采用简洁的语言介绍理论，通过实例训练来介绍 Visual FoxPro 9.0 的基本概念和语法，学生只需基本理解项目实例中各模块的设计思想，初步掌握修改方法，就能以该模块作为开发模板，在其他项目的开发中应用。

本书项目 1、项目 2 和项目 9 由黑龙江工商职业技术学院的梁玉凤老师编写，项目 3 和项目 4 由东北石油大学计算机与信息技术学院的赵娅老师编写，项目 5 和项目 6 由甘肃林业职业技术学院的党进才老师编写，项目 7 和项目 8 由上海行建职业学院的蔡红老师编写，全书由梁玉凤老师统稿。

全书中的例题有详细注释，便于自学。本书还配有电子课件、习题答案、源代码等资源，有需要的读者可以登录华信教育资源网（www.hxedu.com.cn）免费下载。

由于作者的水平有限，书中难免有错误和不妥之处，恳请读者不吝赐教。

编 者

2011 年 7 月

目 录

项目 1 创建学生学籍管理系统项目	1
任务 1.1 熟悉 VFP 基本知识	1
1.1.1 Visual FoxPro 9.0 的安装和启动	2
1.1.2 Visual FoxPro 9.0 的用户界面	3
1.1.3 Visual FoxPro 9.0 的配置	6
1.1.4 常量与变量	8
1.1.5 表达式	12
1.1.6 函数	18
1.1.7 项目管理器的功能和界面	31
任务 1.2 实例训练	34
1.2.1 新建“学生学籍管理系统”项目	34
1.2.2 查看“学生学籍管理系统”的详细内容	35
1.2.3 添加或移去文件	35
1.2.4 创建和修改文件	36
1.2.5 查看表中数据	37
1.2.6 添加文件说明	37
1.2.7 定制项目管理器	37
1.2.8 设置项目管理器字体	39
1.2.9 在项目之间共享文件	39
任务 1.3 项目分析	39
任务 1.4 模仿设计与自主设计	40
1.4.1 VFP 的辅助设计工具	40
1.4.2 VFP 项目管理器的使用	40
项目小结 1	40
项目习题 1	41
项目 2 设计学籍管理数据库	43
任务 2.1 熟悉数据库基础知识	43
2.1.1 数据和信息	44
2.1.2 数据库系统的基本概念	45
2.1.3 数据模型	46
2.1.4 关系数据库	49
任务 2.2 实例训练	52
2.2.1 创建“学籍管理数据库”	52
2.2.2 打开“学籍管理数据库”	54
2.2.3 设置“学籍管理数据库”为当前数据库	55



2.2.4 关闭“学籍管理数据库”	55
2.2.5 向“学籍管理数据库”中添加表	56
2.2.6 修改“学籍管理数据库”	58
2.2.7 删除“学籍管理数据库”	58
任务 2.3 项目分析	58
任务 2.4 模仿设计与自主设计	59
2.4.1 数据库的创建	59
2.4.2 数据库中表的操作	59
项目小结 2	59
项目习题 2	60
项目 3 设计学生学籍管理系统中的表	62
任务 3.1 熟悉表的基础知识	63
3.1.1 表的结构	63
3.1.2 自由表和数据库表	63
3.1.3 修改表结构	64
3.1.4 索引	64
3.1.5 索引的创建	66
3.1.6 索引的使用	67
3.1.7 排序	68
3.1.8 排序与索引的区别	69
3.1.9 数据完整性	69
任务 3.2 实例训练	70
3.2.1 利用表设计器创建“课程表”	70
3.2.2 在“学籍管理数据库”下创建“学生信息表”	72
3.2.3 修改“学生信息表”的结构	77
3.2.4 显示“学生信息表”记录	79
3.2.5 浏览和修改“学生信息表”中的记录	85
3.2.6 向“学生信息表”中追加记录	86
3.2.7 往“学生信息表”中插入记录	89
3.2.8 删除“学生信息表”中的一些记录	89
3.2.9 记录的查询定位	95
3.2.10 在项目管理器中建立索引	98
3.2.11 利用菜单打开索引	100
3.2.12 对“学生信息表”进行排序	101
3.2.13 设置域完整性约束	102
3.2.14 建立各表之间的关系	104
3.2.15 设置参照完整性约束	105
任务 3.3 项目分析	106
3.3.1 学生信息表	106
3.3.2 课程表	106
3.3.3 成绩表	106
任务 3.4 模仿设计与自主设计	107
3.4.1 创建表	107

3.4.2	表的基本操作	108
3.4.3	表中数据的操作	108
3.4.4	创建数据库中多表之间的联系	108
3.4.5	表中字段属性设置	108
3.4.6	记录的定位和筛选	108
项目小结 3	109
项目习题 3	109
项目 4	设计学生学籍管理系统中的查询与视图	112
任务 4.1	熟悉查询与视图基本知识	112
4.1.1	查询	112
4.1.2	视图	113
4.1.3	SQL 概述	113
4.1.4	SQL 查询功能	113
4.1.5	SQL 操作功能	122
4.1.6	SQL 定义功能	123
任务 4.2	实例训练	124
4.2.1	启动查询设计器	124
4.2.2	建立单表查询和多表查询	126
4.2.3	运行查询	129
4.2.4	利用视图设计器创建单表本地视图	130
4.2.5	利用视图设计器创建多表本地视图	133
4.2.6	利用视图向导创建单表本地视图	135
4.2.7	利用本地视图更新数据表中的数据	138
任务 4.3	项目分析	139
4.3.1	学生信息查询	139
4.3.2	学生信息与成绩查询	140
4.3.3	学生信息视图	140
4.3.4	学生信息与成绩视图	141
4.3.5	课程信息视图	142
任务 4.4	模仿设计与自主设计	142
4.4.1	建立查询文件	142
4.4.2	建立视图文件	142
项目小结 4	143
项目习题 4	143
项目 5	学生学籍管理系统程序设计	146
任务 5.1	熟悉程序设计基本知识	147
5.1.1	程序的建立与编辑	147
5.1.2	程序的调用与执行	148
5.1.3	简单的输入/输出命令	148
5.1.4	顺序结构程序	149
5.1.5	选择结构程序	150
5.1.6	循环结构程序	151



5.1.7 过程	153
5.1.8 参数传递及变量的作用域	154
任务 5.2 实例训练	155
5.2.1 顺序结构程序设计实例	155
5.2.2 选择结构程序设计实例	156
5.2.3 循环结构程序设计实例	158
5.2.4 过程调用实例	160
5.2.5 参数传递实例	161
5.2.6 变量的作用域实例	162
任务 5.3 项目分析	163
5.3.1 查询程序	163
5.3.2 成绩等级评定程序	164
5.3.3 修改成绩程序	165
任务 5.4 模仿设计与自主设计	165
5.4.1 闰年程序	165
5.4.2 随机数产生程序	165
5.4.3 素数程序	165
项目小结 5	166
项目习题 5	166
项目 6 设计学生学籍管理系统的表单	169
任务 6.1 熟悉表单基础知识	169
6.1.1 表单设计器环境	169
6.1.2 表单控件的基本操作	172
6.1.3 常用表单控件	173
6.1.4 面向对象的概念	177
6.1.5 Visual FoxPro 基类	179
6.1.6 Visual FoxPro 容器类与控件类	179
6.1.7 常用事件	179
任务 6.2 实例训练	180
6.2.1 利用表单向导创建表单	181
6.2.2 利用表单设计器创建表单	184
6.2.3 表单的修改与运行	185
6.2.4 为表单建立数据环境	185
6.2.5 给表单添加标签控件	187
6.2.6 给表单添加命令按钮控件	189
6.2.7 给表单添加文本框控件	191
6.2.8 给表单添加编辑框控件	193
6.2.9 设计查询统计表单	194
任务 6.3 项目分析	195
6.3.1 “欢迎进入系统”表单	195
6.3.2 “系统登录”表单	195
6.3.3 “学生信息”表单	196
6.3.4 “查询统计”表单	196

任务 6.4 模仿设计与自主设计	196
6.4.1 “ 欢迎使用图书管理系统 ” 表单	196
6.4.2 “ 图书信息 ” 表单	197
6.4.3 “ 借阅者信息查询统计 ” 表单	197
项目小结 6	197
项目习题 6	197
项目 7 设计学生学籍管理系统的菜单	199
任务 7.1 熟悉菜单基础知识	199
7.1.1 菜单系统的结构	200
7.1.2 下拉式菜单设计过程	200
7.1.3 下拉式菜单定义方法	202
7.1.4 快捷菜单设计	204
任务 7.2 实例训练	205
7.2.1 下拉式菜单设计实例	205
7.2.2 快捷菜单设计实例	207
7.2.3 给菜单指定菜单任务	208
7.2.4 运行菜单	209
任务 7.3 项目分析	210
7.3.1 系统主菜单	210
7.3.2 “ 学生管理 ” 菜单	210
7.3.3 “ 编辑 ” 菜单	211
任务 7.4 模仿设计与自主设计	211
7.4.1 建立常规菜单	211
7.4.2 建立快捷菜单	212
7.4.3 菜单综合设计	212
项目小结 7	212
项目习题 7	212
项目 8 设计学生学籍管理系统报表	214
任务 8.1 熟悉报表基础知识	214
8.1.1 报表概述	214
8.1.2 利用报表向导创建报表	215
8.1.3 利用报表设计器创建报表	215
8.1.4 打印报表	216
任务 8.2 实例训练	217
8.2.1 利用报表向导创建报表实例	217
8.2.2 利用报表设计器创建报表实例	219
8.2.3 修改报表	222
任务 8.3 项目分析	222
8.3.1 students 报表	222
8.3.2 学生学籍信息报表	223
8.3.3 学生成绩报表	223



任务 8.4 模仿设计与自主设计	224
8.4.1 创建“图书信息报表”	224
8.4.2 创建快速报表“租借信息报表”	224
项目小结 8	224
项目习题 8	224
项目 9 公司人事管理系统	226
任务 9.1 系统规划	226
9.1.1 模块化设计	226
9.1.2 系统流程图设计	227
9.1.3 详细设计	227
任务 9.2 系统设计	227
9.2.1 创建数据表	227
9.2.2 创建表单	229
9.2.3 创建菜单和报表	234
任务 9.3 系统完善	236
9.3.1 填写项目信息	236
9.3.2 生成应用程序	237
任务 9.4 模仿设计与自主设计	238
项目小结 9	238
项目习题 9	239
项目习题参考答案	240
2010 年 9 月计算机二级 VFP 笔试试题	242
2010 年 9 月计算机二级 VFP 笔试试题参考答案	248
2011 年计算机二级考试 VFP 考试大纲	249
参考文献	252



项目 1

创建学生学籍管理系统项目

项目要点◎

Visual FoxPro 概述。

Visual FoxPro 语言基础，包括常量、变量的类型及各种类型的表达式。

Visual FoxPro 的项目管理器的功能。

项目管理器的操作。

学习目标◎

熟悉 Visual FoxPro 的基础知识。

掌握常量、变量的应用方法。

重点掌握表达式和函数应用。

了解 Visual FoxPro 项目管理器的界面。

掌握创建项目和对项目管理器进行操作的方法。

任务 1.1 熟悉 VFP 基本知识

Visual FoxPro 9.0 是 Microsoft 公司推出的 Visual FoxPro 的最新版本，是一个优秀的可视化数据库编程工具。它在以往版本的基础上有了很大的改进，添加了 XML 处理能力，增强了与外部交换数据的能力，提供了可视化界面的设计方法，支持面向对象的程序设计技术，功能更加强大。Visual FoxPro 9.0 是具备自开发语言的数据库管理系统，它既可以作为



大型数据库的前端开发工具，也可以进行小型的应用系统开发，它将面向对象的程序设计技术与关系型数据库系统有机地结合在一起，是性能完善的程序设计语言，扩展了对应用程序的管理和在 Internet 上发布用户数据的功能，使得用户开发数据库的工具更加完善与快捷，成为一种功能强大且可靠的数据库管理系统。

1.1.1 Visual FoxPro 9.0 的安装和启动

Visual FoxPro 9.0 可以从 CD-ROM 或通过网络安装。本书介绍从 CD-ROM 安装的方法。

1. 安装 Visual FoxPro 9.0

操作步骤如下。

(1) 将 Visual FoxPro 9.0 系统安装光盘插入 CD-ROM 驱动器中，Visual FoxPro 安装开始页面即自动运行。

(2) 单击“Install Visual FoxPro”链接来加载 Visual FoxPro 安装程序。

(3) 为确认是否需要安装附带组件，单击“Prerequisites”按钮来显示所有的必备组件。

(4) 单击“Install Now!”按钮来安装所有的新组件。如果安装 Visual FoxPro 的先决条件所要求的组件只需要进行组件升级，则单击“Update Now”按钮。

(5) 设置完成后可能需要重新启动计算机，单击“Done”按钮，Visual FoxPro 安装界面会重新出现。

(6) 继续安装，单击“Visual FoxPro”。

(7) 在接受最终用户许可协议，并输入产品密钥和用户名后，单击“Continue”按钮。

(8) 在选项页面，选择需要安装的功能部件，单击“Install Now”按钮，继续安装。

(9) 安装完成后，单击“Done”按钮返回 Visual FoxPro 安装界面，单击“Exit”按钮返回 Visual FoxPro 安装起始页面。

若系统中已经装有中文版的 Visual FoxPro 9.0，可通过再次启动安装程序来增加或删除系统构件，或者重新安装，或者全部删除系统。运行安装程序 SETUP.EXE，会再次显示安装程序的界面。

2. 启动 Visual FoxPro 9.0

启动 Visual FoxPro 有多种方法，常用的有以下 3 种。

(1) 从“开始”菜单启动。在“开始”菜单中单击“程序”|Microsoft Visual FoxPro|Visual FoxPro 9.0 命令，启动 Visual FoxPro 9.0 系统。

(2) 从“资源管理器”启动。进入“资源管理器”窗口，在文件夹 VFP9 下找到 VFP9 图标，双击该图标，即可启动 Visual FoxPro 9.0 系统。

(3) 从“运行”对话框启动。在“开始”菜单中单击“运行”命令，弹出“运行”对话框，在该对话框中输入 VFP9.EXE 然后单击“确定”按钮，即可启动 Visual FoxPro 9.0。

3. 退出 Visual FoxPro 9.0

可以用以下 5 种方法退出 Visual FoxPro 9.0 系统。

(1) 在主窗口中用鼠标单击其右上角的“关闭”按钮。

(2) 在命令窗口中输入命令 QUIT，并按 Enter 键。

- (3) 单击“文件”菜单中的“退出”命令。
- (4) 按组合键 Alt+F4。
- (5) 按组合键 Ctrl+Alt+Del，进入“任务管理器”窗口，选择 Visual FoxPro，单击“结束任务”按钮。

1.1.2 Visual FoxPro 9.0 的用户界面

根据上述启动方式，进入到 Visual FoxPro 系统环境的主界面，如图 1-1 所示。

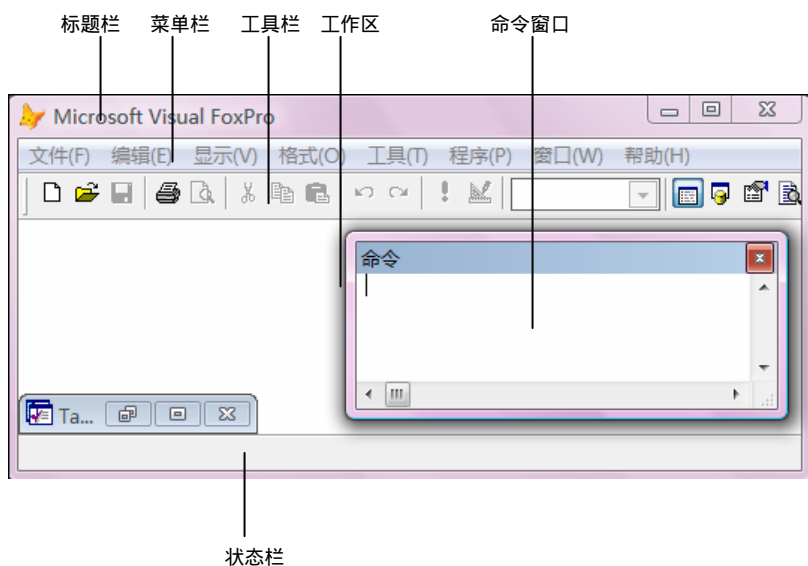


图 1-1 Visual FoxPro 主界面

Visual FoxPro 界面与一般的 Windows 应用程序相似，由标题栏、菜单栏、工具栏、工作区、状态栏和命令窗口等组成，各部分功能如下。

1. 标题栏

标题栏位于系统主窗口的第一行，包含系统程序图标、主窗口标题、最小化按钮、最大化按钮和关闭按钮等对象。

(1) 系统程序图标。单击系统程序图标可打开主窗口的控制菜单，通过控制菜单选项可以控制屏幕的移动和大小。双击该图标可关闭系统。

(2) 主窗口标题。主窗口标题是系统定义的窗口名称，可以用__SCREEN 或__VFP 命令改变其内容。

(3) 最小化按钮。单击最小化按钮，可将 Visual FoxPro 系统主窗口缩小成按钮，并存放在 Windows 桌面底部的任务栏中。若需要再次打开该窗口，可在任务栏中单击 Visual FoxPro 按钮。

(4) 最大化按钮。单击最大化按钮，可将 Visual FoxPro 系统主窗口定义为最大窗口，此时窗口没有边框。

(5) 关闭按钮。单击关闭按钮，可以关闭 Visual FoxPro 系统。

2. 菜单栏

菜单是交互方式下实现人机对话的工具。菜单栏位于主窗口的第二行，根据分类组合，



它包含“文件”、“编辑”、“显示”、“格式”、“工具”、“程序”、“窗口”和“帮助”8个菜单，每个菜单中都有若干菜单命令。大多数菜单命令可以通过菜单方式执行，单击其中一个菜单命令可以执行一个操作。

使用键盘执行命令的方法是：按“Alt+字母键”，可激活一个菜单命令。例如：按Alt+E组合键，则弹出“编辑”下拉菜单，然后通过方向键移动方向光标至所需要的命令，再按Enter键即可执行该命令。

(1)“文件”菜单提供有关系统和文件的基本操作，通过其中的菜单选项可以完成创建、打开和保存文件，以及对文件的其他操作，也可以实现打印机的管理并完成打印功能。

(2)“编辑”菜单可完成对文本或控件的编辑、查找和替换等操作。

(3)“显示”菜单控制报表、标签、表单设计器及工具栏的显示，允许定制操作表的工作方式，菜单中的选项由当前工作环境确定。当用户尚未打开要显示的文件时，“显示”菜单只有一个命令（工具栏）；当用户已打开表、表单或报表等文件时，“显示”菜单的命令将附加一些与打开文件相对应的菜单命令。

(4)“格式”菜单中包含“控制字体格式”、“文本缩进”等选项，该菜单的选项根据打开窗口的不同而不同。

(5)“工具”菜单中包含“向导”菜单命令，以及“编程工具”、“程序调试器”、“系统环境设置”等命令。

(6)“程序”菜单包含与程序编译、运行有关的命令。

(7)“窗口”菜单包含对已打开窗口进行管理的若干命令。

(8)“帮助”菜单提供如何获得 Visual FoxPro 在线帮助的信息，以及有关部门如何得到技术支持的信息。

3. 工具栏

初次启动 Visual FoxPro 9.0 时，常用工具栏位于菜单栏的下面。Visual FoxPro 系统提供的工具栏包含了可完成一般性操作的按钮及经常使用的对象。通过工具栏中的按钮可以方便地完成那些经常性操作。

Visual FoxPro 9.0 提供了不同环境下的 11 种常用的工具栏，即：“常用”工具栏、“布局”工具栏、“表单控件”工具栏、“表单设计器”工具栏、“查询设计器”工具栏、“视图设计器”工具栏、“数据库设计器”工具栏、“报表控件”工具栏、“报表设计器”工具栏、“调色板”工具栏和“打印预览”工具栏。这些工具栏可以帮助用户方便地进行各种操作，用户可用鼠标将其拖动到任意位置。

用户可以根据需要和自己的习惯来定制自己的系统。定制工具栏包括定制工具栏的种类，定制每个工具栏中的项目，定制工具栏的显示方式。

(1)定制工具栏的种类。除了系统默认显示的工具栏外，用户可以根据需要创建或修改自己的工具栏，成为定制工具栏。

操作步骤如下。

在菜单栏中单击“显示”|“工具栏”命令，弹出“工具栏”对话框，如图 1-2 所示（若图中某个选项前面的方格中显示“×”，则表示该工具栏已显示在界面上）。

在“工具栏”对话框中选择需要放到界面上的工具栏。

单击“确定”按钮。

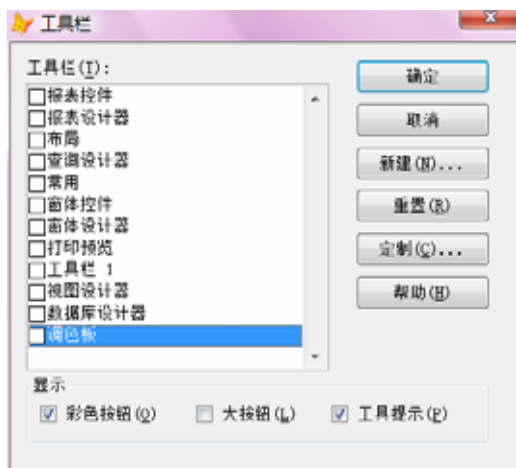


图 1-2 “工具栏”对话框

(2) 定制每个工具栏中的项目。系统允许用户按照自己的习惯重新组织各个工具栏，包括各工具栏按钮的位置、种类等。用户既可以将所需要的任何工具栏按钮放到任何一个工具栏中，也可以增减或删除任何一个工具栏中的任何项目。

操作步骤如下。

在“工具栏”对话框中单击“定制”按钮，弹出“定制工具栏”对话框，如图 1-3 所示。

选择工具栏的类别，对话框右边将显示该工具栏所包含的按钮。单击其中一个按钮，可以显示该按钮的功能。

用鼠标将所需要的按钮拖动到系统工具栏上任意类的任意位置，即可将该按钮加入到相应的工具栏中。对于不再需要的按钮，可以用相反的方法，用鼠标将该按钮拖离工具栏。

如果需要创建新的工具栏，可以在“工具栏”对话框中单击“新建”按钮，弹出“新工具栏”对话框，如图 1-4 所示。

输入工具栏名称后，单击“确定”按钮，将显示新建的工具栏。此时，该工具栏是空的，可以根据需要加入按钮。加入按钮的方法与定制时的方法完全一样（可以从任意工具栏中选取任意按钮加入到新工具栏中）。

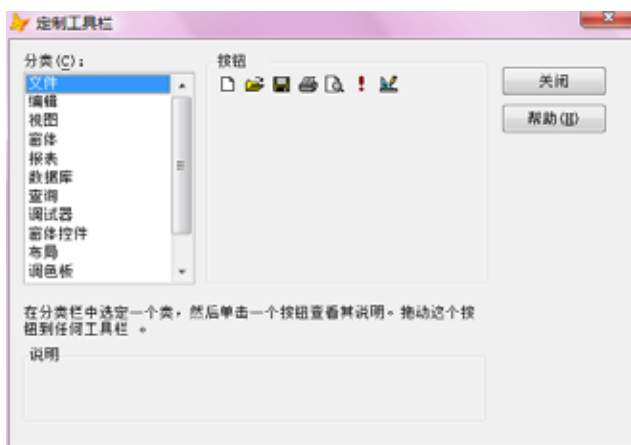


图 1-3 “定制工具栏”对话框

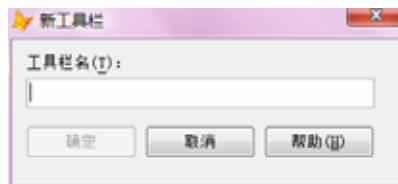


图 1-4 “新工具栏”对话框



(3) 定制工具栏的显示方式。工具栏的显示方式包括固定显示方式和浮动显示方式。固定显示方式是指工具栏始终显示在系统工具栏中；浮动显示方式是指工具栏以一个独立的小窗口形式显示，可以任意拖动，窗口的标题就是该工具栏的标题。

两种方式的转换方法如下。

如果需要将固定显示方式的工具栏变成浮动显示方式，可把鼠标定位在该工具栏中按钮之间的空隙位置，然后拖动该工具栏离开工具栏区域即可。

如果需要将浮动显示方式的工具栏变成固定显示方式，可把鼠标定位在该工具栏小窗口的标题上，然后拖动该工具栏窗口到工具栏区域，出现单条的矩形框时释放鼠标即可。

4. 命令窗口

命令窗口位于工具栏与状态栏之间，是界面上的一个重要部件，是 Visual FoxPro 系统执行命令和编辑的窗口。在命令窗口中，既可以直接输入 Visual FoxPro 9.0 命令，按回车键后立即执行，实现对数据库的操作和管理，也可以用各种编辑工具对操作命令进行修改、插入、删除、剪切、赋值和粘贴等操作，还可以在该窗口中建立并运行命令文件，已经执行的命令会在窗口中自动保留。如果要执行前面已经输入的命令，只要将光标移到该命令的任意位置处按回车键即可。

命令窗口的显示和关闭，有 3 种方法进行控制。

(1) 单击命令窗口右上角的“关闭”按钮可关闭命令窗口。可以通过“窗口”菜单进行控制，在菜单栏中单击“窗口”|“命令窗口”命令，可以打开命令窗口。

(2) 在菜单栏中单击“窗口”|“隐藏”命令，可以关闭窗口。

(3) 单击“常用”工具栏上的“命令窗口”按钮完成显示与隐藏。

5. 工作区

系统工作区是指工具栏与状态栏之间的一大块空白区域，各种工作窗口将在这里展开。

6. 状态栏

状态栏位于界面的最底部，用于显示某一时刻数据库管理系统对数据进行管理的工作状态。如果当前工作区中没有打开的表文件，状态栏的内容为空白；如果当前工作区中有打开的表文件，状态栏将显示表名、表所在的数据库名、表中当前记录的记录号、表中的记录总表里当前记录的共享状态等内容。用户可以通过 SET STATUS ON/OFF 命令设置状态栏的显示与关闭，如果是 SET STATUS OFF 状态，系统不显示状态栏，如果是 SET STATUS ON 状态，系统显示状态栏。

1.1.3 Visual FoxPro 9.0 的配置

Visual FoxPro 的配置除了决定其外观，还决定了系统的操作环境和工作方法。安装完 Visual FoxPro 之后，系统用默认的值来设置环境。为了满足不同用户的要求，可通过系统环境改变其设置。系统设置包括添加或删除 Visual FoxPro 控件，更新 Windows Registry 注册项，改变状态栏和工具栏，安装 ODBC 数据源等。

Visual FoxPro 系统环境可通过系统提供的“选项”对话框，选择相应的选项卡，进行配置。

在 Visual FoxPro 系统主菜单下，选择“工具”|“选项”命令，打开“选项”对话框，如图 1-5 所示。

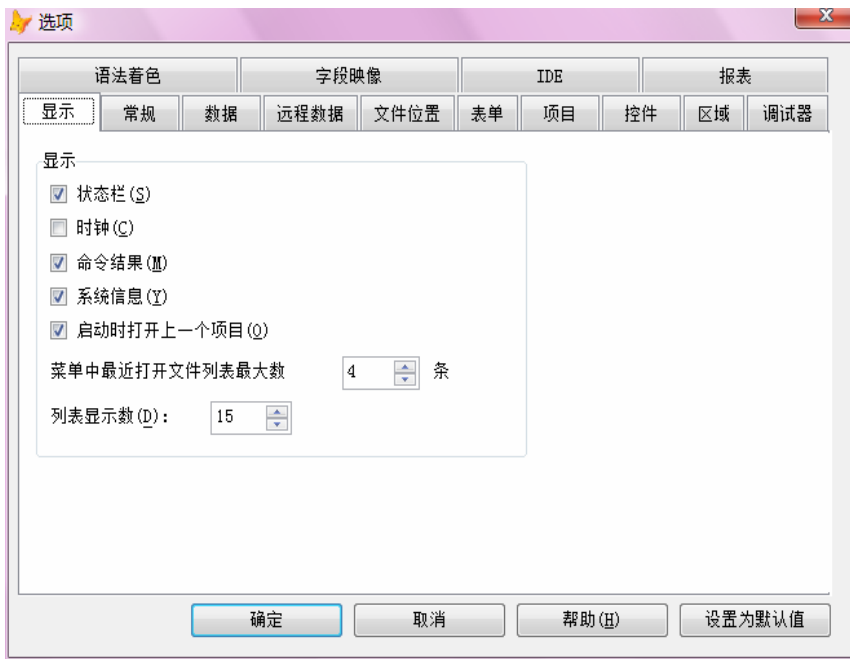


图 1-5 “选项”对话框

系统环境设置的意义是修改系统的默认值，除了使用“选项”对话框进行设置外，还可以通过 SET 命令及配置文件进行设置。“选项”对话框的选项功能如表 1-1 所示。

表 1-1 “选项”对话框的选项功能

选项卡	各选项配置功能
显示	该选项卡的功能为是否显示状态栏、时钟、命令结果或系统信息
常规	该选项卡的功能是数据输入与编程选项，如设置警告声音、是否记录编译错误或自动填充新记录、使用的定位键、调色板使用的颜色、改写文件之前是否警告等
数据	字符串比较设定、表选项，如是否使用 Rushmore 优化、是否使用索引强制唯一性、备注块大小、查找的记录计数器间隔及使用什么锁定选项
远程数据	链接超时限定值、一次拾取记录数目及如何使用 SOL 更新
文件位置	设置 Visual FoxPro 默认目录位置，帮助文件及辅助文件存储在何处
表单	表单设计器选项，确定网格面积、所用的刻度单位、最大设计区域及使用何种模板类
项目	设置是否提示使用向导、双击时运行或修改文件及源代码管理选项
控件	设置是否使用表单控件工具栏中的“查看类”按钮、所提供的可视类库和 ActiveX 控件选项
区域	设置日期、时间、货币及数字的格式
调试	设置调试器显示、跟踪选项及使用什么字体和颜色
编辑器	设置编辑器的选项及程序元素所用的字体及颜色
字段映像	设置从数据环境设计器、数据库设计器或项目管理器向表单拖放表或字段时创建何种控件
IDE	设置文件或传户口的样式及外观与行为
报表	设置数据与输出的默认值、设计器的默认值等信息



1.1.4 常量与变量

1. 常量

常量是指在程序运行过程中其值不会发生改变的数据,以直观的数据形态和意义直接出现在程序中,表示一个具体的、不变的值。不同类型的常量有不同的书写格式,通常有 6 种类型。

(1) 数值型常量。数值型常量 (numeric) 由数字、小数点和正负号组成。例如, 123.98、0.00012、745、-6.78、0 等。为了表示很大或很小的数值型常量,也可以使用科学记数法(浮点型)书写。例如,用 7.65E-12 表示 7.65×10^{-12} ,用 7.65E+12 表示 7.65×10^{12} 。数值型常数的最大数据长度为 20 位(包括+、-号和小数点),其取值范围是 -0.9999999999E+19 ~ 0.9999999999E+20。数值型数据在内存中用 8 个字节表示。

(2) 货币型常量。货币型常量 (currency) 用来表示货币值,货币数据在存储和计算时,采用 4 位小数。如果多于 4 位小数,系统会自动将多余的小数位四舍五入。例如,常量 \$65.789678 将存储为 \$65.7897。

货币型常量没有科学记数形式,在内存中占用 8 个字节,取值范围是 -922 337 203 477.5807 ~ 922 337 203 685 477.5807。货币型常量书写格式与数值型常量类似,但要加上一个前置的符号(\$)。

(3) 字符型常量。字符型常量 (character) 由汉字和 ASCII 字符集中可打印字符组成,包括 26 个英文字母、10 个阿拉伯数字、各种符号空格及其他专用字符,也称为字符串,其表示方法是用单引号、双引号或方括号把字符串括起来。单引号、双引号或方括号称为字符串的定界符。定界符不作为常量本身的内容,但它规定了常量的类型及起始和终止界限。



注意: 字符型常量的定界符必须成对匹配,不能一边用单引号而另一边用双引号。如果某种定界符本身也是字符串的内容,则需要用另一种定界符为该字符串定界。不包含任何字符的字符串 ("") 叫空串。空串与包含空格的字符串 (" ") 不同。

【例 1-1】 单问号和双问号功能示例。

```
? "数据库应用系统", '123', [ABXabc], ["数据"123]
?? "学习", '字符串', [表示方法]
```

单问号 (?) 命令的功能是在下一行显示若干个表达式的值。

双问号 (??) 命令的功能是在同一行显示表达式的值。

(4) 日期型常量。日期型常量 (date) 是表示日期的特殊数据,有固定的格式,一般因国家而异。它的定界符是一对花括号。花括号内包括年、月、日 3 部分内容,各部分内容之间用分隔符分隔。分隔符可以是斜杠 (/)、连字号 (-)、句点 (.) 和空格。例如, {^2008/09/05}、{^2006-09-05}、{^2005.08.05} 和 {^2002 09 12} 都可以,其中斜杠是系统在显示日期型数据时使用的默认分隔符。日期型数据固定为 8 个字符。

(5) 日期时间型常量。日期时间型常量 (datetime) 包括日期和时间两部分内容: {<日期>, <时间>}。<日期>部分与日期型常量相似, <时间>部分的格式为 {hh:mm:ss}, hh、mm 和 ss 分别代表时、分和秒, 默认值分别为 12、0 和 0。AM (或 A) 和 PM (或 P) 分别代表上午和下午, 默认值为 AM。时间部分的取值范围是: 00:00:00 AM ~ 11:59:59 PM。例如, {^2006-09-05 11:12:36 PM}。

(6) 逻辑型常量。逻辑型常量 (logical) 表示逻辑判断结果的值, 只有逻辑真和逻辑假两个值。逻辑真的常量的表示形式有: .T.、.t.、.Y. 和 .y.。逻辑假的常量的表示形式有: .F.、.f.、.N. 和 .n.。前后两个句点作为逻辑型常量的定界符是必不可少的, 否则会被误认为变量名。逻辑型数据只占用 1 字节。

2. 变量

变量是指用标志符命名的用于存放数据值的计算机内存单元。在程序执行的每个瞬间, 变量的值都是确定的, 每个变量在内存中占用一定的存储单元, 其中存放变量的值, 在程序执行过程中会被新值所代替, 是可以改变的。变量可分为字段变量和内存变量两大类。

(1) 字段变量。一个数据库由若干相关的数据表组成, 一个数据表由若干具有相同属性的记录组成, 每个记录又由若干个字段组成。字段变量又称字段名变量, 是指数据表中已定义的任意一个字段。字段变量与记录一起构成数据表的基本结构。字段变量是数据表中最基本的数据单元, 它随着数据库文件的建立而产生, 表中的各条记录对同一个字段名可能取值不同, 因此, 表中的字段名就是字段变量。

使用字段变量必须先建立数据表, 建立数据表时, 应先定义字段变量的字段名称、字段类型和长度。字段变量的定义一般通过表设计器来进行。字段变量的数据类型与该字段定义的类型一致, 其类型有数值型、字符型、逻辑型、日期型、日期时间型、备注型和通用型等。

(2) 内存变量。内存变量是独立于数据库而存在的一种临时内存工作单元, 它通常用来存放程序运行过程中用户输入的各种信息、中间数据、中间结果和最后处理的结果等。内存变量是内存中的一个存储区域, 变量值就是存放在该存储区域中的数据, 其类型取决于变量值的类型。例如, 当把一个常量赋给一个变量时, 该常量就被存放在该变量对应的存储位置中而成为该变量新的取值。在 Visual FoxPro 中, 内存变量的类型可以改变, 也就是说, 可以把不同类型的数据赋给同一个变量。

内存变量的数据类型有: 字符型 (C)、数值型 (N)、货币型 (Y)、逻辑型 (L)、日期型 (D) 和日期时间型 (T)。

每个内存变量必须有一个名称, 以标志它在该内存单元的存储位置, 用户可通过变量标志来存取常量, 也就是说通过变量名访问变量。如果当前表中存在一个同名的字段变量, 则在访问内存变量时, 必须在变量名前加上前缀 M. (或 M.->), 否则系统将访问同名的字段变量。

向内存变量赋值不必事先定义, 变量的赋值命令有以下两种格式。

STORE <表达式> TO <变量名表>

<内存变量名>=<表达式>

功能: 计算表达式表中的值, 然后把计算结果赋给内存变量。



注意：等号一次只能给一个内存变量赋值。

STORE 命令可以同时给若干变量赋予相同的值，各内存变量名之间必须用逗号分开。

不是用内存变量文件保存内存变量的值，退出系统后，内存变量的值会消失。

在 Visual FoxPro 中，一个变量在使用之前并不需要特别地声明或定义。当用 STORE 命令给变量赋值时，如果该变量并不存在，那么系统会自动建立它。

可以通过对内存变量重新赋值来改变其内容和类型。

【例 1-2】 内存变量和字段变量的访问。

这里用到一个表文件“学生信息表.DBF”，其结构如下。

学生（学号 C（8），姓名 C（6），性别 C（2），出生日期 D（8））

在命令窗口输入的命令：

```
USE 学生信息表      &&USE 是打开表的命令
LIST                &&LIST 是显示表中各条记录的命令
```

主屏幕显示的内容为：

记录号	学 号	姓 名	性别	出生日期
1	JX0601	王 磊	男	03/09/70
2	RJ0602	张芳芳	女	12/03/76
3	WH0603	马晓倩	女	11/23/82
4	WL0604	李小莉	女	03/19/80

在命令窗口输入命令：

```
? 学号,姓名,性别,出生日期
```

主屏幕显示的内容为：

```
1  JX0601  王 磊  男  03/09/70
```

在命令窗口输入命令：

```
STORE "王小玉" TO 姓名      &&用字符串向内存变量姓名赋值
XM=姓名                    &&用字段变量赋值内存变量
X1=-32.4                   &&用数值型常量-32.4 赋值内存变量
STORE 12.3 TO  Y1,Y2        &&对两个内存变量 Y1、Y2 赋予相同值
STORE .F. TO  A1            &&用逻辑常量赋值内存变量 Q
? 姓名, M.XM,  Y1,Y2,X1
```

主屏幕显示的内容为：

```
王 磊 王小玉  12.3  12.3  -32.4
```

（3）数组变量。数组变量是一组有序内存变量的集合，它由一系列元素组成，每个内存变量都是这个数组的一个元素，或者说，数组是由同一个名字组织起来的内存变量的集

合，每个数组元素都可通过数组名及相应的下标来访问。每个数组元素相当于一个简单变量，可以为各元素分别赋值。在 Visual FoxPro 中，一个数组中各元素的数据类型可以不同。数组在使用之前一般要用 DIMENSION 或 DECLARE 命令定义，规定数组的维数、数组名和数组大小。数组大小由下标值的上、下限决定。

定义数组的命令格式如下。

DIMENSION/ DECLARE <数组名 1>(<下标 1>[,<下标 2>]),[<数组名 2>(<下标 1,<下标 2>)] ...

功能：定义一个或多维数组及其下标的上界，系统规定各下标的下界为 1。

例如：DIMENSION A(6), M(2, 3)。

一维数组 A 含 6 个元素：A(1)、A(2)、A(3)、A(4)、A(5)、A(6)。

二维数组 M 含 6 个元素：M(1, 1)、M(1, 2)、M(1, 3)、M(2, 1)、M(2, 2)、M(2, 3)。

整个数组的数据类型为 A (Array)，各个数组元素可以分别存放不同类型的数据。在使用数组和数组元素时，应注意以下问题。

凡使用简单内存变量的地方，均可以使用数组元素。

在赋值和输入语句中使用数组名时，表示将同一个值同时赋给该数组的全部数组元素。

在同一个运行环境下，数组名不能与简单变量名重复。

赋值语句中的表达式位置不能出现数组名。

可以用一维数组的形式访问二维数组。例如，数组 A 的各元素用一维数组形式可依次表示为：A(1)、A(2)、A(3)、A(4)、A(5)、A(6)，其中 A(4)与 A(2, 1)是同一变量。

(4) 内存变量的显示。

格式 1：LIST MEMORY [LIKE<通配符>][TO PRINTER][TO FILE<文件名>]

格式 2：DISPLAY MEMORY [LIKE<通配符>][TO PRINTER][TO FILE<文件名>]

功能：在屏幕上显示内存变量的当前信息，包括变量名、属性、类型、内容、定义它们的过程和函数名，同时给出当前内存变量的总数及可再定义的内存变量数目。

选用 LIKE 子句只显示与通配符相匹配的内存变量。通配符包括“*”和“?”，“*”表示任意多个字符，“?”表示任意一个字符。

选用子句 TO PRINTER，用于在显示的同时送往打印机。

TO FILE<文件名>，存入给定文件名的文本文件，文件的扩展名为.txt。

LIST 和 DISPLAY 的功能基本相同。LIST MEMORY 一次显示与通配符匹配的所有内存变量，如果内存变量多，一屏显示不下则自动上下滚动。DISPLAY MEMORY 分屏显示与通配符匹配的内存变量，如果内存变量多，每显示一屏后暂停，按任意键之后再继续显示下一屏。

【例 1-3】 display 功能示例。

DISPLAY MEMORY LIKE A*	&&显示所有名字以 A 开头的内存变量
DISPLAY MEMORY LIKE ?M*	&&显示所有名字中第二个字符为 M 的内存变量

(5) 内存变量的清除。

格式 1：CLEAR MEMORY



格式 2 : RELEASE <内存变量名表>

格式 3 : RELEASE ALL [LIKE <通配符> EXTENDED]

格式 4 : RELEASE ALL [LIKE <通配符> EXCEPT <通配符>]

功能：格式 1 清除所有内存变量；格式 2 清除指定的内存变量；格式 3 清除所有的内存变量，在人机会话状态其作用与格式 1 相同，如果出现在程序中则应该加上短语 EXTENDED，否则不能删除公共内存变量；格式 4 选用 LIKE 子句清除与通配符相匹配的内存变量，选用 EXCEPT 子句清除与通配符不相匹配的内存变量。

【例 1-4】 release 功能示例。

```
RELEASE ALL LIKE F*  &&表示只清除名称以 F 开头的所有内存变量
RELEASE ALL EXCEPT A?  &&表示在现有的内存变量中，除变量名为两个字符并且以 A 开头的变量之外，将其他的内存变量清除。如果该命令出现在程序中，则不能清除公共内存变量
```

1.1.5 表达式

运算是数据加工的过程，描述各种不同运算的符号称为运算符，参与运算的数据称为运算对象，也可称为操作数。

运算对象有字段名、内存变量、数组元素名、常量和函数等。

运算符有算术运算符、关系运算符、字符运算符和逻辑运算符等。

表达式是用运算符将运算对象连接起来所组成的式子，按照规定的法则，每个表达式最终能计算出一个结果并且具有某种类型，该类型称为表达式的类型。

根据表达式值的类型，表达式可分为数值表达式、字符表达式、日期时间表达式和逻辑表达式。大多数逻辑表达式是带比较运算符的关系表达式。

1. 数值表达式

数值表达式由算术运算符连接数值型数据、数值型常量、数值型变量和数值型函数而成，其运算结果仍然是数值型。算术运算符的优先级如表 1-2 所示。

表 1-2 算数运算符的优先级

运 算 符	功 能	优 先 级	说 明
()	括号	最高	形成表达式内的子表达式
**, ^	幂	高	乘方运算
*, /	乘, 除	次之	乘除运算
%	模运算(取余)	较低	求余运算
+, -	加, 减	最低	加减运算

【例 1-5】 计算数学表达式 $(5+3^{(1+2)})/(12-8)$ 的值。

在命令窗口输入命令：

```
? (5+3^ (1+2)) / (12-8)
```

主屏幕显示的内容为：

```
8
```

【例 1-6】 求余运算。

在命令窗口输入命令：

```
? 36%6, 15%4, 15%-4
```

主屏幕显示的内容为：

```
0      3      -1
```

2. 字符表达式

字符表达式由字符运算符、字符型常量、字符型内存变量、字符型数组、字符型字段和字符型函数等组成，其运算结果是字符型或逻辑型数据。字符串运算符有以下两个，它们的优先级相同。

＋：前后两个字符串首尾连接形成一个新的字符串。

－：连接前后两个字符串，并将前字符串的尾部空格移到合并后的新字符串尾部。

当运算符前面字符串尾部没有空格时，两种运算符的运算结果一样。数值型数据只要加定界符就是字符串。

字符表达式的一般格式：

<字符串 1><字符串运算符><字符串 2>

【例 1-7】 字符串运算示例。

在命令窗口输入命令：

```
"Visual "+"FoxPro 9.0"
```

主屏幕显示的内容为：

```
Visual FoxPro 9.0
```

在命令窗口输入命令：

```
"Visual "-"FoxPro 9.0"
```

主屏幕显示的内容为：

```
VisualFoxPro 9.0
```

3. 日期时间表达式

日期时间表达式由日期运算符和日期时间型常量、日期时间型内存变量、数组、日期时间型函数等组成，其运算结果是日期时间型。

日期时间表达式中可以使用的运算符也有“＋”和“－”两个。

运算符“＋”的运算结果是在给定日期上加天数或在给定的时间日期上加秒数。

运算符“－”的运算结果是给定两个日期相差的天数或给定两个日期时间相差的秒数。

运算符“＋”和“－”既可以作为日期时间运算符，也可以作为算术运算符和字符串连接运算符，到底作为哪种运算符使用，要由它们所连接的运算对象的数据类型而定。

日期时间表达式的格式有一定限制，不能任意组合，不能用运算符“＋”将两个日期连接起来。对日期时间表达式格式的规定如表 1-3 所示，其中的<天数>和<秒数>都是表达式。



表 1-3 日期时间运算符格式

格 式	结果及类型
<日期>+<天数>	日期型。指定日期若干天后的日期
<日期>-<天数>	日期型。指定日期若干天前的日期
<日期>-<日期>	数值型。两个指定日期相差的天数
<日期时间>+<秒数>	日期时间型。指定日期时间若干秒后的日期时间
<日期时间>-<秒数>	日期时间型。指定日期时间若干秒前的日期时间
<日期时间>-<日期时间>	数值型。两个指定日期时间相差的秒数

【例 1-8】 日期时间运算示例。

在命令窗口输入命令：

```
? { ^ 1998/01/19 }+10
```

主屏幕显示的内容为：

```
01/29/98
```

在命令窗口输入命令：

```
? { ^ 1998/01/19 }-10
```

主屏幕显示的内容为：

```
01/09/98
```

在命令窗口输入命令：

```
? { ^2001/03/19 }-{ ^2000/03/19 }
```

主屏幕显示的内容为：

```
365
```

在命令窗口输入命令：

```
? { ^2001/03/19 8:40:10 }+100
```

主屏幕显示的内容为：

```
03/19/01 08:41:50 AM
```

在命令窗口输入命令：

```
? { ^2001/09/10 8:41:50 }-{ ^2001/09/10 8:40:10 }
```

主屏幕显示的内容为：

```
100                      &&表示为相差的秒数
```

4. 关系表达式

关系表达式由关系运算符和字符表达式、算术表达式、时间日期表达式组成，其运算结果为逻辑型常量。关系运算是对运算符两边同类型元素进行比较，若关系成立，则结果为.T.，反之，结果为.F.。

关系表达式通常也称为简单逻辑表达式，它由关系运算符将两个运算对象连接起来形成，即：<表达式 1> <关系运算符> <表达式 2>：

关系运算符的作用是对两边两个同类型的表达式进行比较。其运算结果是逻辑型数据。运算符及其含义如表 1-4 所示，它们的优先级相同。

表 1-4 关系运算符

运 算 符	说 明	运 算 符	说 明
<	小于	< >、 }或! :	不等于
>	大于	=	等于
<=	小于等于	\$	子串包含测试
>=	大于等于	==	字符串精确比较

运算符“==”和“\$”仅适用于字符型数据。其他运算符适合于任何类型的数据，但前后两个运算对象的数据类型一致，各种类型数据的比较规则如下。

(1) 数值型数据相比较，按数值的大小比较，包括正负号。

(2) 日期和日期时间型数据比较，将日期看成 YYMMDD 的 6 位数，按数值大小比较，越早的日期或时间越小，越晚的日期或时间越大。

(3) 数字与字母的比较，数字的 ASCII 码均小于字母的 ASCII 码。

(4) 西文字符是按照 ASCII 码值排列的，空格在最前面。

关系运算各类数据比较如下。

数值型数据比较关系运算示例。

【例 1-9】 数值型数据比较示例。

在命令窗口输入命令：

```
? 4<3,-10>-21,4*6=26      &&两个数值型数据相比较
```

主屏幕显示的内容为：

```
.F.      .T.      .F.
```

日期和日期时间型数据比较关系运算示例。

【例 1-10】 日期型数据比较示例。

在命令窗口输入命令：

```
? {^2002/01/10}>{^2001/12/28}  &&两个日期型数据相比较
```

主屏幕显示的内容为：

```
.T.
```

逻辑型数据比较。

【例 1-11】 逻辑型数据比较示例。

在命令窗口输入命令：

```
? .T.>.F.
```

主屏幕显示的内容为：

```
.T.
```



子串包含测试。

一般格式：<前字符型表达式>\$<后字符型表达式>

“\$”包含运算符的功能是比较后面的字符串中是否含有前一个字符串，如有，则结果为.T.，否则结果为.F.。

【例 1-12】关系运算示例。

在命令窗口输入命令：

```
? "张" $ "张斌", "ABCD" $ "ABCD1234", "微型计算机" $ "计算机"
```

主屏幕显示的内容为：

```
.T.           .T.           .F.
```

在命令窗口输入命令：

```
? "李" < "王", "王老师" < "王", "您好" > "你好"    &&汉字比较
```

主屏幕显示的内容为：

```
.T.           .F.           .T.
```

在命令窗口输入命令：

```
? "abc"=="abc", "abc"=="abc"    &&字符串精确比较
```

主屏幕显示的内容为：

```
.T.           .F.
```

5. 逻辑表达式

逻辑表达式由逻辑运算符和逻辑型常量、逻辑内存变量、逻辑型数组、返回逻辑型数据、关系表达式组成，其运算结果为逻辑型常量。逻辑运算符有 3 个：.NOT.或!（逻辑非）、.AND.（逻辑与）及.OR.（逻辑或）。可以省略两端的点，写成 NOT、AND、OR。其优先级顺序依次为 NOT，AND，OR。

逻辑运算符的运算优先级和规则如表 1-5 和表 1-6 所示。

表 1-5 逻辑运算符的优先级

运 算 符	功 能	优 先 级
()	括号	最高
.not.	逻辑非	高
.and.	逻辑与	较低
.or.	逻辑或	最低

表 1-6 逻辑运算符的运算规则

A	B	.not.A	.not.B	A.and.B	A.or.B
.t.	.t.	.f.	.f.	.t.	.t.
.t.	.f.	.f.	.t.	.f.	.t.
.f.	.t.	.t.	.f.	.f.	.t.
.f.	.f.	.t.	.t.	.f.	.f.

【例 1-13】 逻辑运算示例。

在命令窗口输入命令：

```
? 3>2 AND 5+2>2
```

主屏幕显示的内容为：

```
.T.
```

&&运算结果为.T.

在命令窗口输入命令：

```
? .NOT.12*4>5
```

&&运算结果为.F.

主屏幕显示的内容为：

```
.F.
```

在命令窗口输入命令：

```
? 2>5.OR.3*20<20
```

&&运算结果为.F.

主屏幕显示的内容为：

```
.F.
```

【例 1-14】 根据所给条件，列出逻辑表达式。

(1) 在学籍库表中，要查找女学生或团员的条件，可用下列逻辑表达式表示。

性别="女".AND.政治面貌="团员"

(2) 查询基本工资高于 2000 元的讲师和副教授，可用下列条件表达式表示。

基本工资>1800 .AND.职称="讲师" .OR.基本工资>=1800 .AND.职称="副教授"

或者写成：

基本工资>=1800 .AND.(职称="讲师" .OR. 职称="副教授")

6. 表达式运算规则

由上所述，可以总结出组成各类表达式的运算符有算术运算符、字符串运算符、关系运算符和逻辑运算符等，在每类运算符中，各个运算符有一定的运算优先级，而不同类型的运算符也可能出现在同一个表达式中，这时它们的运算优先级如下。

第一级，算术运算符和字符串运算符。

算术运算符包括正负号、**或^、*、/、-、+。

字符串运算符级别相同，包括+、-。

第二级，关系运算符。

关系运算符级别相同，包括<、<=、=、>=、>、<>或#、\$、==。

第三级，逻辑运算符。

逻辑运算符包括 NOT 或！、AND、OR。

所有同一级运算都是从左至右进行的，括号内的运算优先执行，嵌在最内层的括号里的运算首先进行，然后由内向外依次执行。圆括号作为运算符，可以改变其他运算符的运算顺序，圆括号中的内容作为整个表达式的子表达式，优先级最高。圆括号可以嵌套。有



时,在表达式的适当地方插入圆括号,并不是为了改变其他运算符的运算次序,而是为了提高代码的可读性。

1.1.6 函数

函数是系统为实现一般表达式不能或难以实现的某种特定操作而编制好的一组程序,供用户调用。它是用程序来实现的一种数据运算或转换,往往需要若干个自变量,即运算对象,但只能有一个运算结果,称为函数值,可以用函数名加一对圆括号加以调用,自变量放在圆括号里。

1. 函数的基本格式和调用方法

(1) 函数的格式。

函数名([<自变量表>])

符号“[]”中内容是可选项,符号“<>”中的内容是必选项,但都不包括这些符号本身。上述格式说明,每个函数都有一个确定的函数名称,圆括号内是函数的自变量,也是参数。自变量在某些函数中是默认的,可以有一个或多个自变量,当出现多个自变量时,用逗号分隔。

(2) 函数的调用。函数可以作为运算对象出现在表达式中。在调用时要注意函数中自变量的个数、顺序和数据类型等。

本章将常用函数分为数值函数、字符处理函数、日期类函数、数据类型转换函数和测试函数5类,通过举例分别介绍。

2. 数值函数

数值函数是指函数值为数值的一类函数,它们的自变量和返回值往往都是数值型数据。

(1) 求绝对值函数。

格式:ABS(<数值表达式>)

功能:ABS()指定的数值表达式的绝对值。

【例 1-15】 求绝对值函数功能示例。

在命令窗口输入命令:

```
? ABS(45) ,ABS(-45) , ABS(100-45) &&显示绝对值函数值
```

主屏幕显示的内容为:

45	45	55
----	----	----

(2) 求整数函数。

格式:INT(<数值表达式>)

CEILING(<数值表达式>)

FLOOR(<数值表达式>)

功能:INT()返回指定数值表达式的整数部分。

CEILING()返回大于或等于指定数值表达式的最小整数。

FLOOR()返回小于或等于指定数值表达式的最大整数。

【例 1-16】 求整数函数功能示例。

在命令窗口输入命令：

```
? INT(-10.5),INT(8.25*2)           &&显示取整函数值
```

主屏幕显示的内容为：

```
-10           16
```

在命令窗口输入命令：

```
? CEILING(4.6),CEILING(-4.6)       &&显示取整函数值
```

主屏幕显示的内容为：

```
5           -4
```

在命令窗口输入命令：

```
? FLOOR(4.6),FLOOR(-4.6)          &&显示取整函数值
```

主屏幕显示的内容为：

```
4           -5
```

(3) 求平方根函数。

格式：SQRT(<数值表达式>)

功能：返回指定表达式的平方根。自变量表达式的值不能为负。

【例 1-17】 求平方根函数功能示例。

在命令窗口输入命令：

```
? SQRT(16),SQRT(2*SQRT(2)),(2*6)^(1/2)  &&显示平方根函数值
```

主屏幕显示的内容为：

```
4.00           1.68           3.46
```

(4) 四舍五入函数。

格式：ROUND(<数值表达式 1>,<数值表达式 2>)

功能：返回指定表达式在指定位置四舍五入后的结果。四舍五入后保留的小数位数由<数值表达式 2>决定。若<数值表达式 2>大于等于 0，那么它表示的是要保留的小数位数；若<数值表达式 2>小于 0，那么它表示的是整数部分的舍入位数，即小数点左边的数，被舍（变零）的位数为<数值表达式 2>绝对值。

【例 1-18】 四舍五入函数功能示例。

在命令窗口输入命令：

```
X=1234.1992  
? ROUND(X,3),ROUND(X,1),ROUND(X,0), ROUND(X,-1)
```

主屏幕显示的内容为：

```
1234.199       1234.2       1234       1230
```



(5) 求余数函数。

格式：MOD(<数值表达式 1>,<数值表达式 2>)

功能：计算<数值表达式 1>除以<数值表达式 2>后所得的余数，<数值表达式 2>是除数，<数值表达式 1>是被除数，若被除数与除数同号，则函数值为<数值表达式 1>除以<数值表达式 2>后所得的余数；如果被除数与除数异号，则函数值为两数相除的余数再加上除数的值，余数的正负号与除数相同。

【例 1-19】 求余数函数功能示例。

在命令窗口输入命令：

```
?MOD(10,3),MOD(-10,-3),MOD(-10,3),MOD(10,-3)
```

主屏幕显示的内容为：

1	-1	2	-2
---	----	---	----

(6) 求最大值和最小值函数。

格式：MAX(<数值表达式 1>,<数值表达式 2>[,<数值表达式 3>...])

MIN(<数值表达式 1>,<数值表达式 2>[,<数值表达式 3>...])

功能：MAX()计算各自变量表达式的值，并返回表达式列表中的最大值。

MIN()计算各自变量表达式的值，并返回表达式列表中的最小值。

自变量表达式的类型可以是数值型、字符型、货币型和日期时间型等，但所有表达式的类型必须相同。

【例 1-20】 求最值函数功能示例。

在命令窗口输入命令：

```
? MAX(54,39,40), ? MAX("2","12","05")
```

主屏幕显示的内容为：

54	2
----	---

3. 字符函数

字符函数是指自变量一般是字符型数据的函数。

(1) 截取子串函数。

格式：SUBSTR(<字符表达式 1>,<数值表达式 1>[,<数值表达式 2>])

功能：从指定表达式值的指定起始位置取指定长度的子串作为函数值。起始位置由<数值表达式 1>决定，长度由<数值表达式 2>决定，若<数值表达式 2>默认，则截取的子串从指定起始位置一直取到最后一个字符。若<数值表达式 1>的值为零，则函数值为一个空串。

【例 1-21】 截取子串函数功能示例。

在命令窗口输入命令：

```
?SUBSTR("ABCDEFGHIJK",5,6)      &&显示截取子串函数值
```

主屏幕显示的内容为：

EFGHIJ

在命令窗口输入命令：

```
?SUBSTR("ABCDEFGHIJK",5)      &&显示截取子串函数值
```

主屏幕显示的内容为：

```
EFGHIJK
```

(2) 从字符左边截取子串函数。

格式：LEFT(<字符表达式>,<长度>)

功能：LEFT()从指定表达式值的左端取一个指定长度的子串作为函数值。

【例 1-22】 从左边截取子串函数功能示例。

在命令窗口输入命令：

```
?LEFT("GOOD STUDENT",4),LEFT("GOOD STUDENT",7) &&显示截取左串函数值
```

主屏幕显示的内容为：

```
GOOD          GOOD  ST
```

(3) 从字符右边截取子串函数。

格式：RIGHT(<字符表达式>,<长度>)

功能：RIGHT()从指定表达式值的右端取一个指定长度的子串作为函数值。

【例 1-23】 从右边截取子串函数功能示例。

在命令窗口输入命令：

```
?RIGHT("GOOD STUDENT",4),RIGHT("GOOD STUDENT",7) &&截取右串函数值
```

主屏幕显示的内容为：

```
DENT          STUDENT
```

(4) 求字符串长度函数。

格式：LEN(<字符表达式>)

功能：LEN()返回指定字符表达式值的长度，即所包含的字符个数，函数值为数值型。

【例 1-24】 求字符串长度函数功能示例。

在命令窗口输入命令：

```
x="中文 Visual FoxPro 9.0"      &&显示字符串长度函数值  
? LEN(x)
```

主屏幕显示的内容为：

```
21
```

(5) 求子串位置函数。

格式：AT(<字符表达式 1>,<字符表达式 2>[,<数值表达式>])

ATC(<字符表达式 1>,<字符表达式 2>[,<数值表达式>])

功能：AT()的函数值为数值型。如果<字符表达式 1>是<字符表达式 2>的子串，则<字符表达式 1>值为首字符在<字符表达式 2>值中的位置；若不是子串，则返回 0。

ATC()与 AT()功能类似，但在子串比较时不区分字母大小写。



第三个自变量<数值表达式>用于表明要在<字符表达式 2>值中搜索<字符表达式 1>第几次出现,其默认值是1。

【例 1-25】 求子串位置函数功能示例。

在命令窗口输入命令:

```
STORE "Visual FoxPro 9.0" TO x
? AT("fox",x),ATC("fox",X),AT("FOX",X,3)
```

主屏幕显示的内容为:

```
0      8      0
```

(6) 删除前后空格函数。

格式: TRIM(<字符表达式>)

LTRIM(<字符表达式>)

ALLTRIM(<字符表达式>)

功能: TRIM()返回指定字符表达式值去掉尾部空格后形成的字符串。

LTRIM()返回指定字符表达式值去掉前导空格后形成的字符串。

ALLTRIM()返回指定字符表达式值去掉前导和尾部空格后形成的字符串。

【例 1-26】 删除空格函数功能示例。

在命令窗口输入命令:

```
? TRIM(" Visual ")+("FoxPro 9.0")      &&消除尾部空格
```

主屏幕显示的内容为:

```
VisualFoxPro 9.0
```

在命令窗口输入命令:

```
? LTRIM("Visual ")+("FoxPro 9.0")      &&消除前置空格
```

主屏幕显示的内容为:

```
VisualFoxPro 9.0
```

在命令窗口输入命令:

```
? "数据库"+ ALLTRIM(" Visual ")+("FoxPro 9.0") &&消除前置、尾部空格
```

主屏幕显示的内容为:

```
数据库 VisualFoxPro 9.0
```

(7) 子串替换函数。

格式: STUFF(<字符表达式 1>,<起始位置>,<长度>,<字符表达式 2>)

功能: 用<字符表达式 2>值替换<字符表达式 1>中由<起始位置>和<长度>指明的子串。替换和被替换的字符个数不一定相等,如果<长度>值是0,则<字符表达式 2>插到<起始位置>指定的字符前面,如果<字符表达式 2>值是空串,那么<字符表达式 1>中由<起始位置>和<长度>指明的子串被删去。

【例 1-27】 子串替换函数功能示例。

在命令窗口输入命令：

```
X="GOODBYE!"  
Y="MORNING"  
? STUFF(X,5,3,Y),STUFF(X,1,4,Y)
```

主屏幕显示的内容为：

```
GOODMORNING!      MORNINGBYE
```

(8) 字符串匹配函数。

格式：LIKE(<字符表达式 1>,<字符表达式 2>)

功能：比较两个字符串对应位置上的字符，若所有对应字符都相匹配，函数返回逻辑真(.T.)，否则返回逻辑假(.F.)。

<字符表达式 1>中可以包含通配符“*”和“?”。“*”可与任何数目的字符相匹配，“?”可以与单个字符相匹配。

【例 1-28】 字符串匹配函数功能示例。

在命令窗口输入命令：

```
? LIKE("?b? ","ab"),LIKE("ab","abcdef"),LIKE("a*","abcd")
```

主屏幕显示的内容为：

```
.F.   .F.   .T.
```

4. 日期和时间函数

Visual FoxPro 系统提供了比较丰富的日期和时间函数，可以方便地对日期型数据进行有关操作，以及进行字符型数据和日期型数据间的转换。

日期和时间函数的自变量一般是日期型数据或日期时间型数据。

(1) 系统日期函数。

格式：DATE()

功能：DATE()返回当前系统日期，函数值为日期型。

【例 1-29】 系统日期函数功能示例。

在命令窗口输入命令：

```
? DATE()          &&显示系统日期
```

主屏幕显示的内容为：

```
08/25/06
```

(2) 系统时间函数。

格式：TIME()

功能：TIME()以 24 小时制、hh:mm:ss 格式返回当前系统时间。

说明：时间按一天 24 小时形式的 8 个字符输出，函数值为字符型。

**【例 1-30】 系统时间函数功能示例。**

在命令窗口输入命令：

```
? TIME ( )
```

 &&显示系统时间

主屏幕显示的内容为：

```
17:39:20
```

(3) 日期时间函数。

格式：DATETIME()

功能：DATETIME()返回当前系统日期时间，函数值为日期时间型。

【例 1-31】 日期时间函数功能示例。

在命令窗口输入命令：

```
? DATETIME ( )
```

 &&显示系统日期时间

主屏幕显示的内容为：

```
08/25/06 04:59:05 PM
```

(4) 求年份函数。

格式：YEAR(<日期表达式> | <日期时间表达式>)

功能：YEAR()从指定的日期表达式或日期时间表达式中返回年份。

【例 1-32】 求年份函数功能示例。

假若当前日期为 2008 年 12 月 2 日。

在命令窗口输入命令：

```
? YEAR ( DATE ( ) )
```

 &&显示年

主屏幕显示的内容为：

```
2008
```

(5) 求月份函数。

格式：MONTH(<日期表达式> / <日期时间表达式>)

功能：MONTH()从指定的日期表达式或日期时间表达式中返回月份。

【例 1-33】 求月份函数功能示例。

假若当前日期为 2008 年 10 月 9 日。

在命令窗口输入命令：

```
? MONTH ( { ^2008/10/09 } )
```

 &&显示月份

主屏幕显示的内容为：

```
10
```

(6) 求天数函数。

格式：DAY(<日期表达式> | <日期时间表达式>)

功能：DAY()从指定的日期表达式或日期时间表达式中返回月中的天数。

【例 1-34】 求天数函数功能示例。

假若当前日期为 2008 年 10 月 9 日。

在命令窗口输入命令：

```
? DAY({^2008/10/09})
```

 &&显示日期

主屏幕显示的内容为：

```
9
```

5. 数据类型转换函数

数据类型转换函数的功能是将某种类型的数据转换成另一种类型的数据。

(1) 将字符串转换成数值的函数。

格式：VAL(<字符表达式>)

功能：将指定的数字字符串转换成一个数值型数据。

说明：任何非数字字符的值均为零，但忽略前置空格，函数值的小数位数由 SET DECIMAL TO 命令决定，系统默认值为 2。

【例 1-35】 数据类型转换函数功能示例。

在命令窗口输入命令：

```
? VAL("12")+VAL("15"),VAL("123.456")
```

主屏幕显示的内容为：

```
27.00      123.46
```

(2) 数值转换成字符串的函数。

格式：STR(<数值表达式>[,<长度>,<小数位数>])

功能：将<数值表达式>的值转换成字符串，函数值为字符型。

说明如下。

转换时根据需要自动进行四舍五入。其中<长度>是返回字符串的总长度，即字符串的长度应该是<数值表达式>值的整数部分位数加上<小数位数>值，再加小数点位，包括 STR 输出的空格。

如果<长度>值小于<数值表达式>的整数位，则数值转换成整数位，忽略小数部分，字符串加前导空格以满足规定的<长度>要求；自动进行四舍五入。

如果<长度>值大于或等于<数值表达式>的整数部分位数（包括负数），则优先满足整数部分，并调整小数的位数，自动进行四舍五入。

如果输出的字符串长度和小数位数均未指定，则函数值为四舍五入取整的数值构成的字符串，默认的字符串长度为 10。

如果指定字符串长度小于<数值表达式>值的整数位，将输出由“*”号组成的信息。

【例 1-36】 数值与字符串转换函数功能示例。

若 X=1234.567，依次写出下述函数的结果。

在命令窗口输入命令：

```
? STR(X,9,2), STR(X), STR(X,5,1), STR(X,3,1)
```




主屏幕显示的内容为：

```
1234.57      1235      1235      ***
```

(3) 字符串转换成日期的函数。

格式：CTOD(<字符表达式>)

功能：CTOD()将<字符表达式>值转换成日期型数据。

【例 1-37】 字符串转换成日期的函数功能示例。

在命令窗口输入命令：

```
? CTOD("08/23/2008")      &&显示字符串转换日期型数据
```

主屏幕显示的内容为：

```
08/23/08
```

(4) 日期转换成字符串。

格式：DTOC(<日期表达式> | <日期时间表达式>[,1])

功能：DTOC()将日期型数据或日期时间数据的日期部分转换成字符串。如果使用选项“1”，字符串的格式总是为 YYYYMMDD，共 8 个字符。

【例 1-38】 日期转换成字符串函数功能示例。

在命令窗口输入命令：

```
? DTOC({^1998/10/01})      &&显示日期转换字符串型数据
```

主屏幕显示的内容为：

```
10/01/98
```

在命令窗口输入命令：

```
? DTOC({^1998/10/01},1)
```

主屏幕显示的内容为：

```
19981001
```

(5) 宏替换函数。

格式：&<字符型变量>[.]

功能：替换出字符型变量的内容，即“&”的值是变量中的字符串。如果该函数与其后的字符无明确分界，则要用“.”作函数结束标志。宏替换可以嵌套使用。

【例 1-39】 宏替换函数功能示例。

在命令窗口输入命令：

```
STORE "XJK" TO DA
USE&DA      &&相当于 USE XJK
XM="王磊"
? &XM"你好!"      &&相当于?王磊"你好!"
```

主屏幕显示的内容为：

```
李小明你好!      &&XM 的值是王磊
```

6. 测试函数

测试函数主要用来测试 Visual FoxPro 系统环境下各种工作进行的状况,了解操作对象的状态,根据测试结果,决定下一步的处理方法或程序走向。

(1) 空值测试函数。

格式: ISNULL(<表达式>)

功能:判断一个表达式的运算结果是否为 NULL 值,若是 NULL 值,返回逻辑真(.T.),否则返回逻辑假(.F.)

【例 1-40】 空值测试函数功能示例。

在命令窗口输入命令:

```
STORE "WEER" TO X
STORE NULL TO Y
?Y,ISNULL(Y), ISNULL(X)
```

主屏幕显示的内容为:

```
.NULL. .T. .F.
```

(2) 数据类型测试函数。

格式: VARTYPE(<表达式>[,<逻辑表达式>])

功能:测试<表达式>的类型,返回一个大写字母,函数值为字符型。字母的含义如表 1-7 所示。

表 1-7 数据类型函数的返回数据类型

返回的字母	数据类型	返回的字母	数据类型
C	字符型或备注型	G	通用型
N	数值型、整型、浮点型或双精度型	D	日期型
Y	货币型	T	日期时间型
L	逻辑型	X	NULL 值
O	对象型	U	未定义

若<表达式>是一个数组,则根据第一个数组元素的类型返回字符串;若<表达式>的运算结果是 NULL 值,则根据<逻辑表达式>值决定是否返回<表达式>的类型。如果<逻辑表达式>值为.T.,就返回<表达式>的原数据类型;如果<逻辑表达式>值为.F.或默认,则返回 X 以表明<表达式>的运算结果是 NULL 值。

【例 1-41】 数据库测试函数功能示例。

在命令窗口输入命令:

```
A1="XXXX"
STORE 10 TO Y
STORE NULL TO X
STORE $100.2 TO Z
VARTYPE(A1),VARTYPE(X),VARTYPE(Y),VARTYPE(Z)
```

主屏幕显示的内容为:

```
C          X          N          Y
```



(3) “空”值测试函数。

格式：EMPTY(<表达式>)

功能：根据指定表达式的运算结果是否为“空”值，返回逻辑真 (.T.) 或逻辑假 (.F.)。

首先要注意，这里所指的“空”值与 NULL 值是两个不同的概念，函数 EMPTY(NULL.) 的返回值为逻辑假 (.F.)。其次，该函数自变量表达式的类型除了可以是数值型之外，还有字符型、逻辑型和日期型等类型。不同类型数据的“空”值，有不同的规定，如表 1-8 所示。

表 1-8 不同类型数据的“空”值规定

数据类型	“空”值	数据类型	“空”值
数值型	0	逻辑型	..F
字符型	空串、空格、制表符、回车、换行	日期型	空 (如 CTOD(""))
货币型	0	日期时间	空 (如 CTOT(""))

(4) 表文件首测试函数。

Visual FoxPro 为每个打开的表设置了一个内部使用的记录指针，指向正在被操作的记录，该记录称为当前记录。记录指针的作用是标志表的当前记录。

表文件的逻辑结构如表 1-9 所示。最上面的记录是首记录，记为 TOP；最下面的记录是尾记录，记为 BOTTOM。第一个记录之前有一个文件起始标志，称为 Beginning of File (BOF)；最后一个记录的后面有一个文件结束标志，称为 End of File (EOF)。使用测试函数能够得到指针的位置。刚打开表时，记录指针总是指向首记录。

表 1-9 表文件的逻辑结构

记录指针	首记录 (TOP)
	第 2 个记录
	第 3 个记录
	...
	第 i 个记录
	...
	尾记录 (BOTTOM)

格式：BOF([<工作区号> | <表别名>])

功能：测试记录指针是否指向当前表文件或指定表文件的文件首，即表文件首是指第一条记录的前面位置，若是返回逻辑真 (.T.)，否则返回逻辑假 (.F.)。

若指定工作区上没有打开表文件，函数返回逻辑假 (.F.)；若表文件中不包含任何记录，函数返回逻辑真 (.T.)。

【例 1-42】表文件首测试函数功能示例。

在命令窗口输入命令：

```
USE XJK
? RECNO(), BOF()
```

主屏幕显示的内容为：

```
1          .F.
```

在命令窗口输入命令：

```
SKIP -1  
? RECNO(),BOF()
```

主屏幕显示的内容为：

```
1 .T.
```

(5) 测试文件尾函数。

格式：EOF([<工作区号> | <表别名>])

功能：测试记录指针是否指向当前表文件或指定表文件的文件尾，即最后一个记录已被读到，若是，则返回逻辑真(.T.)，否则返回逻辑假(.F.)。表文件尾是指最后一条记录的后面位置。若默认自变量，则测试当前表文件。

此函数适合于按顺序方式读表文件中最后一个逻辑记录，即测试记录指针是否达到了文件尾。若指定工作区上没有打开表文件，函数返回逻辑假(.F.)；若表文件中不包含任何记录，函数返回逻辑真(.T.)。

【例 1-43】 测试文件为函数功能示例。

在命令窗口输入命令：

```
USE XJK  
GO BOTTOM  
? EOF()
```

主屏幕显示的内容为：

```
.F.
```

在命令窗口输入命令：

```
SKIP  
? EOF()
```

主屏幕显示的内容为：

```
.T.
```

(6) 记录号测试函数。

格式：RECNO([<工作区号> | <表别名>])

功能：返回记录指针指向当前表文件（若默认自变量）或指定表文件中的记录号。如果指定工作区上没有打开表文件，则函数值为 0；如果记录指针指向数值，则为表文件中的记录数加 1；如果记录指针指向文件首，则函数值为表文件中的记录号。

【例 1-44】 记录号测试函数功能示例。

在命令窗口输入命令：

```
USE XJK  
? RECNO()
```

主屏幕显示的内容为：

```
1
```



在命令窗口输入命令：

```
SKIP 2
? RECNO()
```

主屏幕显示的内容为：

```
3
```

(7) 记录删除测试函数。

格式：DELETED([<表的别名> | <工作区号>])

功能：测试指定的表中或在指定工作区中打开的表，看记录指针所指的当前记录是否有删除标记“*”，若有则为真，否则为假，若默认自变量，则测试当前工作区中打开的表。

【例 1-45】记录删除测试函数功能示例。

在命令窗口输入命令：

```
USE XSK
DELETE FOR 性别="男".AND.专业="计算机网络"
LIST 姓名 性别 FOR DELETED()
```

主屏幕显示的内容为：

记录号#	学号	姓 名	性别
4	* WL0604	李小莉	女
5	* WL0605	张国明	男
6	* WL0606	付会军	男

(8) 条件测试函数。

格式：IF(<逻辑表达式>,<表达式 1>,<表达式 2>)

功能：测试<逻辑表达式>的值，若为逻辑真(.T.)，函数返回<表达式 1>的值；若为逻辑假(.F.)，函数返回<表达式 2>的值。<表达式 1>和<表达式 2>的类型不要求相同。

【例 1-46】条件测试函数功能示例。

在命令窗口输入命令：

```
X=10                                &&给 X 赋值
Y=50                                &&给 Y 赋值
? IIF(X>10,X-5,X+50),IIF(Y>10,Y-10,Y+50)  &&根据条件显示结果
```

主屏幕显示的内容为：

```
60                                40
```

(9) 值域测试函数。

格式：BETWEEN(<表达式 T>,<表达式 L>,<表达式 H>)

功能：判断一个表达式的值是否介于另外两个表达式的值之间。当<表达式 T>的值大于或等于<表达式 L>的值且小于或等于<表达式 H>的值时，函数值为逻辑真(.T.)，否则函数值为逻辑假(.F.)。如果<表达式 L>和<表达式 H>中有一个是 NULL 值，那么函数值也是 NULL 值。

该函数的自变量类型可以是数值型、字符型、日期型、日期时间型、逻辑型和货币型

等，但 3 个自变量的数据类型要一致。

【例 1-47】 值域测试函数功能示例。

在命令窗口输入命令：

```
STORE .NULL. TO X           &&给 X 赋空值
STORE 100 TO Y               &&给 Y 赋数值型
? BETWEEN(150,Y,Y+200),BETWEEN(90,X,Y)  &&显示区域函数的结果
```

主屏幕显示的内容为：

```
.T.                .NULL.
```

在命令窗口输入命令：

```
USE XJK
LIST FOR BETWEEN(出生日期,{01/01/70},{12/31/76})&&显示区域函数内日期型
数据的结果
```

以上是 Visual FoxPro 9.0 提供的一些主要函数，在此仅举出一些常用函数的例子，希望读者在今后的深入学习中逐渐体会各种函数的功能。

1.1.7 项目管理器的功能和界面

开发一个应用程序常常需要使用许多文件，除了程序文件外，还有存放数据的表文件、管理表的数据库文件、查询数据的查询文件、制作报表的报表文件、显示和编辑数据的表单文件和组织程序功能的菜单文件等。Visual FoxPro 提供了一个非常有用的工具：项目管理器。项目管理器是用于跟踪创建程序所需要的所有程序、表单、菜单、表、标签、查询和一些其他类型的文件的工具。在建立表、数据库、查询、表单、报表和创建应用程序时，可用来它来组织和管理文件。

Visual FoxPro 常用的文件扩展名及其对应的文件类型如表 1-10 所示。

表 1-10 Visual FoxPro 常用的文件扩展名及其对应的文件类型

文件类型	扩展名	文件类型	扩展名
表文件	.DBF	表备注文件	.FPT
报表文件	.FRM	报表备注文件	.FRM
标签文件	.LBX	标签备注文件	.LBT
表单文件	.SCX	表单备注文件	.SCT
数据库文件	.DBC	复合索引文件	.CDX
项目文件	.PJX	项目备注文件	.PJT
文本文件	.TXT	屏幕格式文件	.FMT
查询程序文件	.QPR	程序文件	.PRG
应用程序文件	.APP	可执行文件	.EXE
菜单格式文件	.MNX	菜单备注文件	.MNT
菜单程序文件	.MPR	内存变量文件	.MEM

项目管理器是 Visual FoxPro 中处理数据和对象的主要工具。项目管理器既可以组织处理表、表单、数据库、报表、查询和其他文件，也可用于管理表和数据库及创建应用程序。项目管理器的界面如图 1-6 所示。

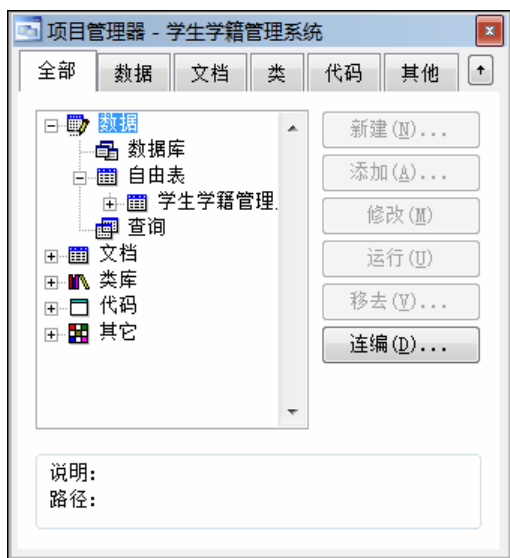


图 1-6 项目管理器的界面

1. 项目管理器的功能

项目管理器窗口由 6 个选项卡、6 个命令按钮和 1 个列表框组成。项目管理器可以完成以下工作。

(1) 在建立表、数据库、查询、表单、报表和应用程序时，可以用项目管理器来组织和管理文件。

(2) 创建一个新项目时，可以将原来已存在文件、表单或 Visual FoxPro 的.dbf 文件添加到新项目中，还可以快速访问 Visual FoxPro 的各种设计器。

2. 项目管理器的按钮

(1) “新建”按钮：创建一个新文件或对象。该按钮与“项目”菜单中“新建文件”选项的作用相同。新建文件或对象的类型与当前选定项的类型相同。

(2) “添加”按钮：把已有的文件添加到项目中。该按钮与“项目”菜单中“添加文件”选项作用相同。

(3) “修改”按钮：在合适的设计器中打开选定项。该按钮与“项目”菜单中“修改文件”选项作用相同。

(4) “运行”按钮：执行选定的查询、表单或程序。当选定项目管理器中的一个查询或表单程序时可用。该按钮与“项目”菜单中“运行文件”选项作用相同。

(5) “移去”按钮：从项目中移去选定的文件或对象。系统将询问是仅从项目中移去该文件，还是同时将其从磁盘中删除。该按钮与“项目”菜单中“移去文件”选项作用相同。

(6) “连编”按钮：连编一个项目或应用程序。该按钮与“项目”菜单中“连编”选项作用相同。

(7) “关闭”按钮：关闭一个打开的数据库。该按钮与“项目”菜单中“关闭文件”选项作用相同，当且仅当选定一个表时可用。如果选定的数据库已关闭，该按钮变为“打开”。

(8) “打开”按钮：打开一个数据库。该按钮与“项目”菜单中“打开文件”选项作用

相同，当选定一个表时可用。如果选定的数据库已打开，该按钮变为“关闭”。

(9)“浏览”按钮：在浏览窗口中打开一个表，该按钮与“项目”菜单中“浏览文件”选项相同，仅在选定一个表时可用。

3. 项目管理器的选项卡

在项目管理器中，可以通过 6 个选项卡来完成各种操作。

(1)“数据”选项卡。可以组织和管理项目文件中包含的所有数据，如数据库、数据表和查询等。

“数据库”选项：数据库是表的集合，一般通过公共字段彼此关联。

“自由表”选项：自由表存储在以.dbf 为扩展名的文件中，它不是数据库的组成部分，如果需要可以将自由表加入数据库中。

“查询”选项：查询是检查存储在表中的特定信息的一种结构化方法，利用查询设计器，可以设置查询的格式，该查询将按照输入的规则从表中提取记录，查询被保存为以.qpr 为扩展名的文件。

(2)“文档”选项卡。可以组织和管理项目文件中利用数据进行操作的文件，如表单、报表和标签等。

“表单”选项：表单主要用于显示和编辑表的内容，可以使用表单设计器设计表单，从而实现对数据的管理。

“报表”选项：报表是一种文件，它告诉 Visual FoxPro 如何设置查询，如何从表中提取结果，以及如何将它们打印出来。

“标签”选项：标签是打印在专用纸上的带有特殊格式的报表。

(3)“类库”选项卡。主要用于显示和管理类库文件。在 Visual FoxPro 中，常见的控件对象有文本框、列表框、复选框、命令按钮和 OLE 控件等，系统允许用户修改或组合这些控件对象，形成一个新的类。这样，用户不仅可以使 Visual FoxPro 提供的控件对象，而且可以重复使用自己新建的类。

(4)“代码”选项卡。主要用于显示和管理 Visual FoxPro 的各类程序代码，包括程序、API 库和应用程序。

“程序”选项：用户用 Visual FoxPro 程序语言编写的最基本的程序代码，扩展名为.prg。

“API 库”选项：指用其他程序设计语言（如 Visual C++）编写的动态链接函数库，Visual FoxPro 程序可以直接调用这些库函数。

“应用程序”选项：指在 Visual FoxPro 环境中设计运行的应用程序，也可以通过编译生成.exe 文件在其他环境中运行。

(5)“其他”选项卡。包括菜单、文本文件和其他文件 3 种类型。

“菜单”选项：利用 Visual FoxPro 提供的菜单设计功能来集成自己的应用程序，设计自己的菜单栏及快捷菜单，然后通过菜单栏及快捷菜单执行相应的命令。

“文本文件”选项：可以用来存储文件的说明信息。

“其他文件”选项：用来存入一些图形文件，如.bmp、.jpg 格式的文件等。

任务 1.2 实例训练

1.2.1 新建“学生学籍管理系统”项目

每开始一个新的任务时，可以创建一个新的项目，然后在项目管理器中创建相应的文件，或者逐步将各种文件添加到项目中。

操作步骤如下。

(1) 单击“文件”菜单下的“新建”命令。

(2) 弹出如图 1-7 所示的“新建”对话框，选中“项目”单选按钮。

(3) 若单击“新建”按钮，则弹出“创建”对话框，用户可在项目名称文本框中输入中文名，我们以“学生学籍管理系统”为例，项目文件以默认的.pjx 扩展名保存，单击“保存”按钮即可。

(4) 若单击“向导”按钮，系统将弹出一个如图 1-8 所示的“应用程序向导”对话框，用户可以指定系统创建一个完整的应用程序，或者只创建一个应用程序结构。



图 1-7 “新建”对话框

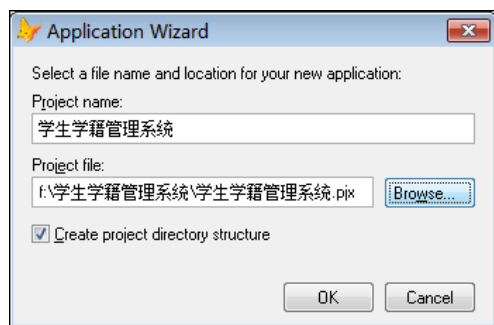


图 1-8 “应用程序向导”对话框

如果选择创建一个完整的应用程序，可以在应用程序中包含以前创建的数据库、表单或报表，或者根据草图使用数据库模板创建一个新的应用程序；如果选择创建一个应用程序结构，以后可以再向结构中添加组件。另外，也可以为项目命名，并指定应用程序或结构文件的存放目录。如果指定的目录不存在，则“应用程序向导”会创建该目录。

(5) 在项目名称文本框中输入“学生学籍管理系统”，并单击“保存”按钮，系统将自动生成该项目管理器的各种应用程序生成器，如图 1-9 所示。

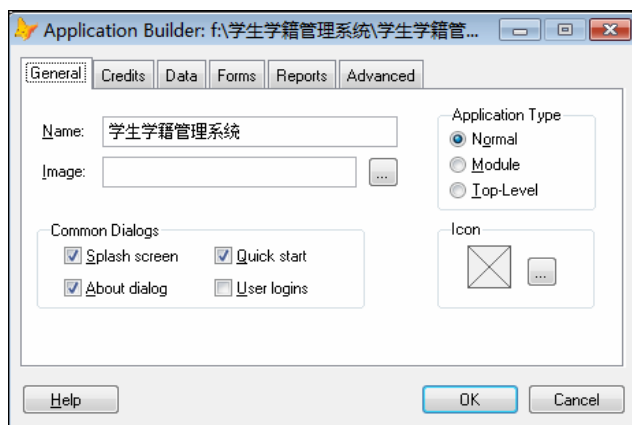


图 1-9 应用程序生成器

(6) 不管创建的是一个完整的应用程序还是结构,“应用程序向导”都会为将来的应用程序创建一个目录和项目结构。

1.2.2 查看“学生学籍管理系统”的详细内容

项目管理器为数据提供了一个组织良好的分层结构图,如图 1-10 所示,它类似于资源管理器的组织结构,操作人员可以将其展开或折叠,以便查看不同层次的详细内容。

如果项目中某个类型的内容不为空,其类型符号旁边会有一个“+”号。单击“+”号,可以看到该项目更详细的内容。若要折叠已展开的列表,可单击列表旁边的“-”号。

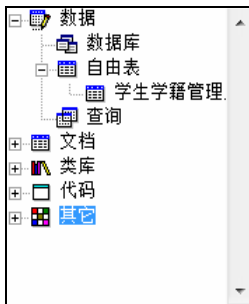


图 1-10 项目管理器分层结构图

1.2.3 添加或移去文件

可以在项目管理器中添加已有的文件,也可以创建新的文件。

(1) 在项目中添加文件。

操作步骤如下。

选择要添加项目的类型,单击“添加”按钮,在弹出的“打开”对话框中选择要添加的文件,然后单击“确定”按钮。

例如,若要把已有的扩展名为.dbf 的表添加到项目中,可在“数据”选项卡中选择“自由表”选项,然后单击“添加”按钮,在弹出的对话框中找到相应的文件,即可将其添加到项目中。



(2) 从项目中移去文件。

操作步骤如下。

选定要移去的内容，在提示框中单击“移去”按钮。如果需从磁盘中删除文件，应单击“删除”按钮。



注意：从项目中“移去”一个文件，只是使该文件不再属于当前项目，但并不是将该文件从系统中删除，该文件仍然存在，需要时还可以重新添加到项目中。从项目中“删除”一个文件，不仅是使该文件不再属于当前项目，而且是将该文件从系统中删除。完成该项操作后，不能再在系统中找到该文件，也无法再将该文件添加到项目中。执行删除操作时要谨慎。

1.2.4 创建和修改文件

在项目管理器中选定要创建或修改的文件类型，单击“新建”或“修改”按钮，系统将显示与所选文件类型相应的设计工具。

(1) 创建一个文件并添加到项目中。

操作步骤如下。

选择要创建的文件类型，单击“新建”按钮。例如，要创建一个“学籍管理数据库”，单击“新建”按钮，弹出“新建数据库”对话框，如图 1-11 所示，选择“新建数据库”，在数据库名后面的文本框中输入“学籍管理数据库”，单击“保存”即可，如图 1-12 所示，同时还可以修改新建的数据库的存储位置。

(2) 修改文件。

操作步骤如下。

选定一个已有的文件，单击“修改”按钮。

例如，要修改一个表，先选定表的名称，然后单击“修改”按钮，该表就显示在表设计器中。



图 1-11 “新建数据库”对话框

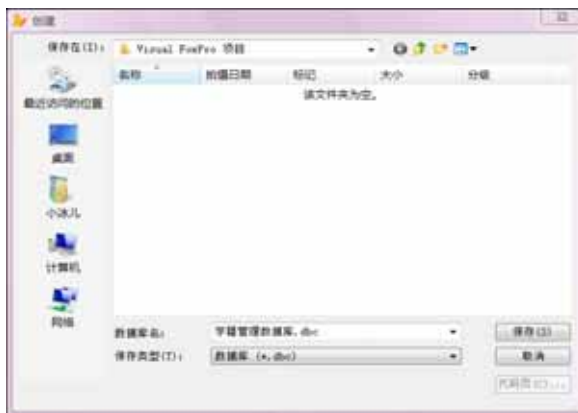


图 1-12 数据库保存对话框

1.2.5 查看表中数据

操作步骤如下。

选择“数据”选项卡，选定一个表并单击“浏览”按钮。例如，我们首先建立自由表 users，选择“数据”选项卡，选择“自由表”，单击“新建”按钮，输入表名“users”，单击“保存”按钮，在弹出的表设计器中输入表中的字段名，并设置类型、宽度等，单击“确定”。选定自由表 users，单击“浏览”按钮，就可以查看该表的情况，如图 1-13 所示，此时表中没有记录。

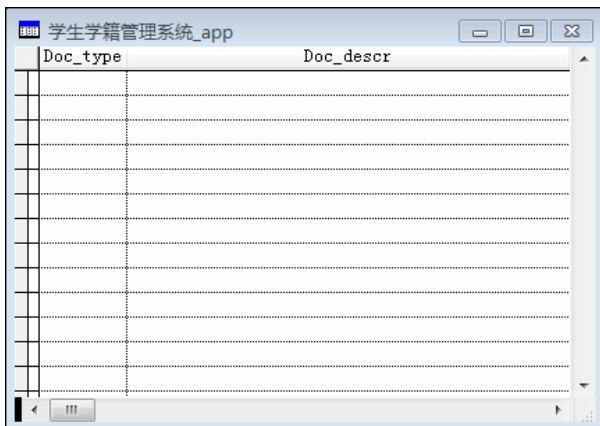


图 1-13 浏览表操作

1.2.6 添加文件说明

创建或添加新的文件时，可以为文件添加说明。以后选定文件时，这些说明将显示在项目管理器的底部。

操作步骤如下。

在项目管理器中选定需要添加说明的文件，在菜单栏上选择“项目”|“编辑说明”选项，在弹出的“说明”对话框中输入文件的说明，单击“确认”按钮。例如，我们为“学生学籍管理系统”添加说明“2011 年 3 月起”，如图 1-14 所示。

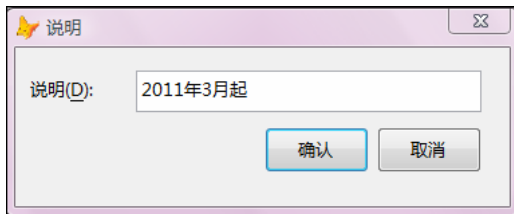


图 1-14 添加文件说明操作

1.2.7 定制项目管理器

项目管理器显示为一个独立的窗口，用户可以移动窗口的位置，改变窗口的尺寸，或者将其折叠起来，只显示选项卡。



(1) 移动项目管理器。

操作步骤如下。

将鼠标指针指向项目管理器的标题栏，然后拖到屏幕上的其他位置。

(2) 改变项目管理器窗口的大小。

操作步骤如下。

将鼠标指针指向项目管理器窗口的顶端、底端、两边或角上，拖动鼠标，即可放大或缩小项目管理器窗口的尺寸。

(3) 折叠/还原项目管理器。

操作步骤如下。

单击项目管理器右上角的向上箭头，即可将项目管理器折叠，只显示选项卡，如图 1-15 所示。此时，向上的箭头变为向下的箭头，单击向下的箭头即可还原。

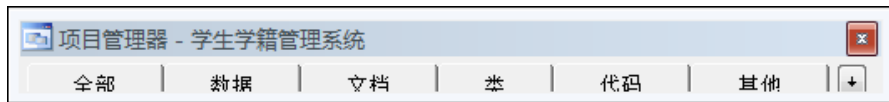


图 1-15 折叠后的项目管理器

(4) 拖开某个选项卡。

操作步骤如下。

折叠项目管理器，从中选定一个选项卡并将其拖离，根据需要重新安排它们的位置。某一选项卡被拖开后，可以在 Visual FoxPro 的主窗口中独立移动，如图 1-16 所示。

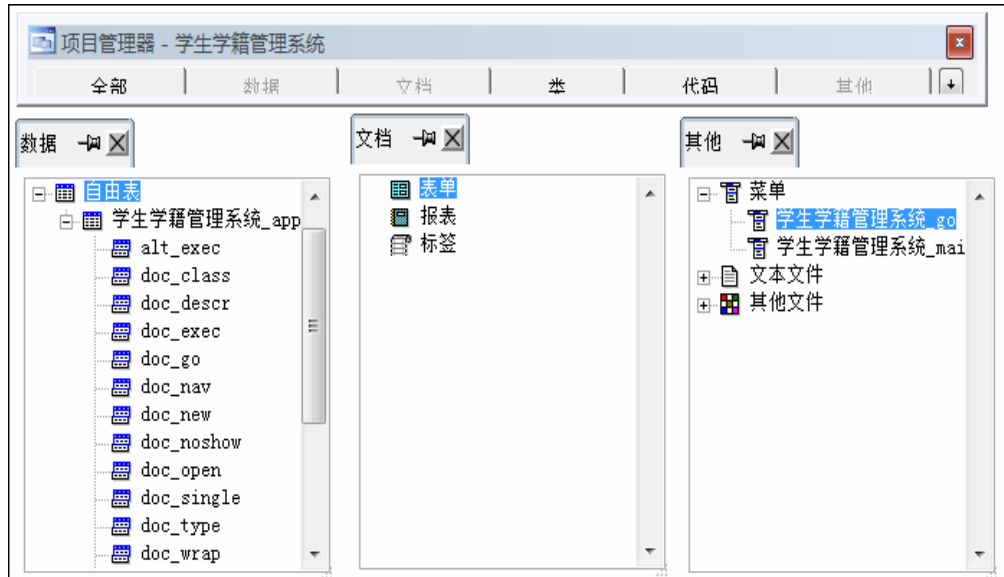


图 1-16 项目管理器选项卡的拖离

当选项卡处于浮动状态时，单击鼠标右键，在弹出的快捷菜单中选择“项目信息”功能，即可访问项目、文件和服务程序中的有关内容，也可以对这些内容进行编辑。

单击选项卡上的“关闭”按钮，或者将选项卡拖回项目管理器，即可还原选项卡。

只要用鼠标将管理器拖至 Visual FoxPro 工具栏的位置，就可以让项目管理器停在 Visual FoxPro 主窗口的顶部，使它像工具栏一样固定在某个位置。此时，项目管理器成为窗口工具栏的一部分，不能将其展开，但可以选择选项卡进行相应的操作。对于停放的项目管理器，同样可以从中拖开选项卡。

1.2.8 设置项目管理器字体

在项目管理器空白处（树状层次结构以外）单击鼠标右键并选择“字体”选项，在打开的“字体”对话框中选择字体类型及大小，即可改变项目管理器窗口中文字的字体，如图 1-17 所示。

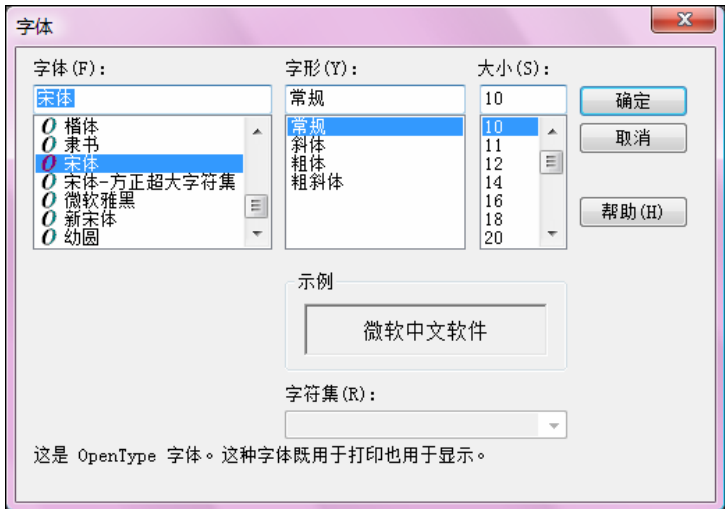


图 1-17 “字体”对话框

1.2.9 在项目之间共享文件

通过与其他项目共享文件，可以使用其他项目的工作成果。此时，项目只存储了对该文件的引用，并未复制该文件。一个文件可同时属于不同的项目。

操作步骤如下。

在系统中打开要共享文件的两个项目，在包含该文件的项目管理器中选择该文件，将文件拖动到另一个项目管理器中。

任务 1.3 项目分析

建立项目，是使用 Visual FoxPro 开发应用程序的第一步，项目创建以后，我们可以通过项目管理器来管理整个项目，同时，我们可以在设置的项目存储目录中，看到系统自动为我们创建的子目录。例如，我们在前面建立的“学生学籍管理系统”的目录结构，如图 1-18 所示。

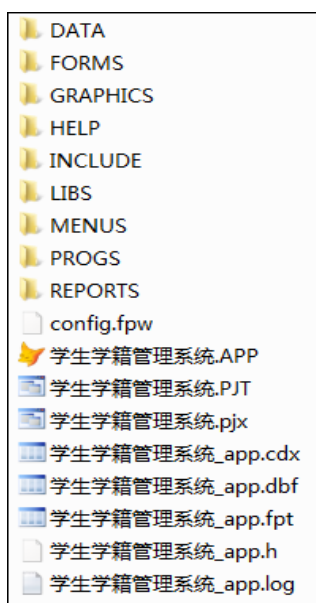


图 1-18 “学生学籍管理系统”的目录结构

任务 1.4 模仿设计与自主设计

1.4.1 VFP 的辅助设计工具

- (1) 在 F 盘建立以自己的姓名命名的文件夹 (如 F:\HHL)。
- (2) 把 F:\HHL 设置为 VFP 的工作目录。

1.4.2 VFP 项目管理器的使用

- (1) 利用系统的“文件”菜单创建一个项目, 名称为“图书管理系统”。
- (2) 打开项目管理器, 了解该项目的结构。

项目小结 1

项目 1 重点介绍了常量、变量、表达式和函数 4 种形式的数据, 以及 Visual FoxPro 的初步知识。其中, 常量和变量是数据运算和处理的基本对象, 表达式和函数体现了语言对数据进行运算和处理的能力, 读者应重点掌握。同时, 项目 1 还比较全面地介绍了 Visual FoxPro 项目管理器的界面, 如何创建项目, 以及对项目管理器进行操作的方法, 这些都是学习 Visual FoxPro 的基础。

项目习题 1

1. 选择题

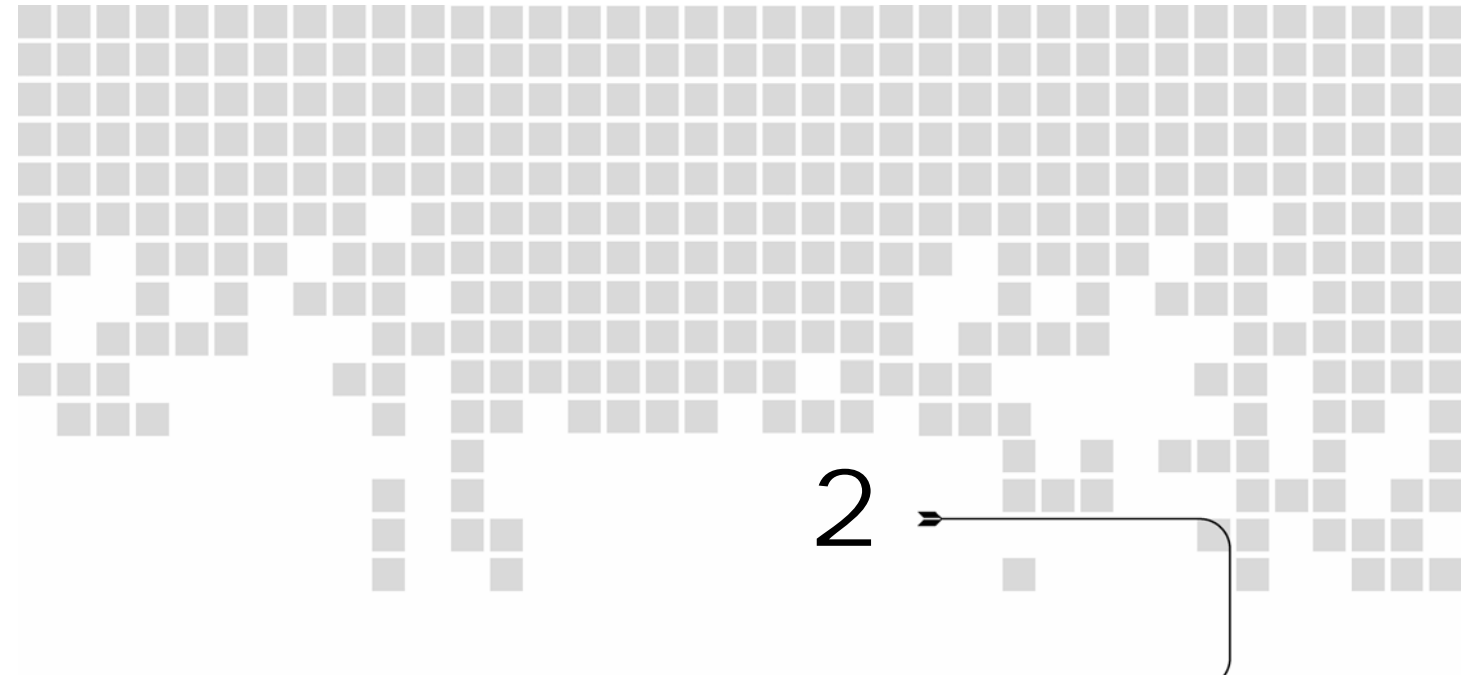
- (1) 下列正确的字符型常数是 ()。
- A. "ABCD"123EFG B. "ABCD'123'EFG"
- C. "ABCD123"EFG D. "ABCD'123"EFG"
- (2) 设 X="102", Y="200", 其值为逻辑假的是 ()。
- A. X\$Y.AND.(X==Y).OR.("X"\$"SJKXY")
- B. X+Y\$"10220089".AND..NOT.X<>Y
- C. .NOT.X\$Y.AND..NOT.X=Y
- D. .NOT.Y-X="98"
- (3) 下列表达式的结果为逻辑值真的是 ()。
- A. "ABC"\$"ACB" B. "ABC"\$"ABXYZPK"
- C. 'ABXYZPK'\$"ABC" D. [AB]\$"ABCDEFGH"
- (4) 表达式('ABC'#'EFG')的运算结果是 ()。
- A. "ABCEFG" B. "EFGABC" C. .F. D. .T.
- (5) 执行下列命令：
- SET EXACT OFF
- ?家='家庭'
- ?关系='关'
- 先后输出的结果为 ()。
- A. .T.、.F. B. .F.、.F. C. .F.、.T.
- (6) 在 Visual FoxPro 9.0 中，下列表达式结果总是逻辑值的是 ()。
- A. 字符运算表达式 B. 算术运算表达式
- C. 关系运算表达式 D. 日期运算表达式
- (7) 退出 Visual FoxPro 9.0 系统的操作方法是 ()。
- A. 单击主窗口右上角的关闭按钮
- B. 在命令窗口中输入 Quit 命令
- C. 按快捷键 Alt+F4
- D. 以上方法都可以
- (8) Visual FoxPro 9.0 中，存储图像的字段类型应该是 ()。
- A. 备注型 B. 通用型 C. 字符型 D. 双精度型
- (9) 连续执行以下命令，屏幕最后显示的结果为 ()。
- SET EXACT OFF
- X="A"
- ?IIF("A"=X,X-"BCD",X+"BCD")
- A. A B. BCD C. ABCD D. A BCD



- (10) 项目管理器中的“全部”选项卡用于显示和管理 ()。
- A. Visual FoxPro 包含的各类文件, 包括数据、文档、类库、代码和其他
B. 数据库、自由表、查询
C. 表单、报表、标签
D. 菜单、文本文件、其他文件
- (11) 项目管理器中的“文档”选项卡用于显示和管理 ()。
- A. 数据、文档、类库、代码、其他
B. 数据库、自由表、查询
C. 表单、报表、标签
D. 程序、API 库、应用程序
- (12) 扩展名为.pjx 的文件表示 ()。
- A. 表文件
B. 备份文件
C. 数据库文件
D. 项目文件
- (13) 下列在项目管理器中移去数据库文件的操作方法, 正确的是 ()。
- A. 选定文件, 单击“项目”菜单中的“移去文件”命令
B. 选定文件, 单击项目管理器上的“移去”按钮
C. 选定文件, 按 Delete 键
D. A, B, C
- (14) 在项目管理器中, 如果某个文件前面出现加号标志, 则表示 ()。
- A. 该文件中只有一个数据项
B. 该文件中有一个或多个数据项
C. 该文件中有多个数据项
D. 该文件不可用
- (15) 函数 APILibraries 在项目管理器的 () 选项卡下。
- A. 数据
B. 文档
C. 代码
D. 其他

2. 填空题

- (1) 表达式{^2005-1-3 10:00:00}-{^2005-10-3 09:00:00}的数据类型是_____。
- (2) 表示“1962 年 10 月 27 日”的日期常量应写为_____。
- (3) 连续执行以下命令, 屏幕最后显示的结果为_____。
- ```
X=STR(12.4,4,1)
Y=RIGHT(X,3)
Z="&Y+&X"
? Z,&Z
```
- (4) 退出 Visual FoxPro 可以在命令窗口输入\_\_\_\_\_命令。
- (5) 项目管理器文件的扩展名是\_\_\_\_\_。
- (6) 项目管理器的“移去”按钮有两个功能, 一个是把文件\_\_\_\_\_, 二是\_\_\_\_\_文件。



# 项目 2

## 设计学籍管理数据库

### 项目要点

- 数据库的基本概念。
- 关系数据库。
- 数据库的基本操作方法。

### 学习目标

- 理解并掌握数据库的基础知识，包括数据、信息、数据管理、数据模型和与数据库系统相关的一些基础理论知识。
- 理解并掌握关系数据模型的相关知识。
- 掌握数据库的基本操作方法。

数据库技术是数据管理的最新成果，被广泛应用于各行各业，为计算机的应用开辟了广阔的天地。从某种意义上说，数据库管理系统是信息处理和数据处理的核心，是研究数据共享的科学，是计算机技术与信息时代相结合的产物。

### 任务 2.1 熟悉数据库基础知识

Visual FoxPro 是历时多年从 dBase、FoxBase 和 FoxPro 发展而来的。通常我们所说的学生信息表、成绩表、课程表和工资表等都是二维数据表，在之前的 Fox 版本中，二维数据表的列项表示字段，行项表示记录，二维数据表是扩展名为.dbf 的数据库文件，这些数据库文件是彼此孤立的，没有一个完整的数据库概念和管理方法。当发展到 Visual FoxPro



时,这种数据库就不叫数据库了,而叫“数据表”,简称“表”。数据库作为一个逻辑上的概念和手段,通过一组系统文件将这些数据表组织在一起进行管理,使得数据表成为相互关联的数据的集合。

在建立 Visual FoxPro 数据库时,相应的数据库名称实际上是扩展名为.dbc 的文件名,并同时自动建立一个扩展名为.dcx 的数据库索引文件和一个扩展名为.dct 的数据库备注文件,即建立一个数据库后,用户可以在磁盘上看到文件名相同,但扩展名分别为.dbc、.dcx 和.dct 的 3 个文件。这 3 个文件是 Visual FoxPro 管理数据库使用的,用户一般不能直接使用这 3 个文件。

### 2.1.1 数据和信息

#### 1. 信息

信息 (information) 是客观事物属性的反映。它所反映的是某一客观系统中,某一事物在某一方面的属性或某一时刻的表现形式。通俗地讲,信息是经过加工处理并对人类客观行为产生影响的数据表现形式。

信息是人们在从事社会活动、经济活动及生产活动时的产物,并用于参与、指导其活动过程。信息是有价值的,信息是可以被感知的。信息可以通过载体传递,也可以通过信息处理工具进行存储、加工、传播、再生和增值。

信息是客观事物属性的反映,是经过加工处理并对人类社会实践和生产经营活动产生重要影响的数据表现形式。或者说,信息是对原始数据加工后得到的,对于某个目的有用的知识或数据。数据是信息的原始资料,数据经过解释并赋予一定的意义后,便成为信息。

在信息社会中,信息一般可与物质和能量相提并论,它是一种重要的资源。

#### 2. 数据

数据 (data) 是反映客观事物属性的记录,是信息的载体。对客观事物属性的记录是用一定的符号来表达的,可见数据是信息的具体表现形式。数据所反映的事物属性是它的内容,而符号是它的形式。从计算机的角度看,数据泛指可以被计算机接受并能够被计算机处理的符号。数据表现信息的形式可以是多种多样的,不仅有数字、文字符号,还有图形、图像和声音等。用数据记录同一信息可以有不同的形式,信息不会随着数据形式的不同而改变其内容和价值。

数据与信息在概念上是有区别的。从信息处理的角度看,任何事物的属性都是通过数据来表示的,数据经过加工处理后具有知识性,并对人类活动产生决策作用,从而形成信息。它们之间的关系也可以看作是原料和成品之间的关系,不经过加工的数据只是一种原始材料,不能影响人们的社会实践和生产经营活动,这种数据只能记录客观世界的事实。只有经过提炼和加工,数据才能发生质的变化,才能成为信息。显然,信息来源于数据,是对数据进行加工处理的产物。数据经过加工后,被赋予一定的含义,具有了知识性并对人类活动产生决策作用,从而形成信息。经过加工的信息,仍然以数据的形式出现,此时的数据是信息的载体,是供人们认识和利用信息的一种媒介。

用数据符号表示信息的形式有许多种,常见的类型有 3 种:一是数值型数据,即对客观事物进行定量记录的符号,如身高、体重、数量、年龄、价格和度数等;二是字符型数

据，即对客观事物进行定性记录的符号，如姓名、单位名称、地址和性别等；三是特殊型数据，即对客观事物进行形象特征和过程记录的符号，如声音、视频和图像等。

总之，信息是有用的数据，数据是信息的表现形式。信息是通过数据符号来传播的，如果数据不具有知识性和有用性，则不能称其为信息。

### 3. 数据处理

数据处理也被称为信息处理，实际上就是利用计算机对各种类型的数据进行处理。它包括对数据的采集、整理、存储、分类、排序、检索、维护、加工、统计和传输等一系列操作。数据处理的目的是从大量的、原始的数据中获得我们所需要的资料并提取有用的数据成分，作为行为和决策的依据。

随着电子计算机软件和硬件技术的发展，数据处理过程发生了划时代的变革，而数据库技术的发展，又使数据处理跨入了一个崭新的阶段。

#### 2.1.2 数据库系统的基本概念

##### 1. 数据库

数据库是指以一定方式存储在计算机存储设备上的相互关联的数据的集合，供数据处理时使用。通俗地讲，数据库就是有条理、有组织、合理地存放大量数据的“数据仓库”。

##### 2. 数据库管理系统 ( DataBase Management System , DBMS )

数据库管理系统是数据库系统的核心，它通过数据库管理软件对所有的数据实行统一规划管理，形成一个数据中心，构成一个“数据仓库”，该数据库中的数据能够满足不同用户的要求，供不同用户共享。在这一管理方式下，应用程序不再只是与一个孤立的数据文件相对应，而是可以取整体数据集的某个子集作为逻辑文件与其对应，通过数据库管理系统实现逻辑文件与物理数据之间的协调与互动，从而实现数据处理。在数据库系统环境下，应用程序对数据的管理和访问灵活方便，而且数据与应用程序之间完全独立，这使程序的编制质量和效率都有所提高。因为数据库文件中的各数据子集间可以建立关联关系，所以数据的冗余大大减少，数据共享性显著增强。

##### 3. 数据库系统 ( DataBase System , DBS )

数据库系统通常是指带有数据库的计算机应用系统，不仅包括数据库本身，即实际存储在计算机中的数据，还包括相应的软件、硬件和各类人员。数据库系统是一个由使用人员、维护人员、加工设备和数据资源构成的完整的计算机应用系统。

一般情况下，数据库系统由相关的硬件环境、软件系统、数据和人员 4 个部分组成。

(1) 硬件环境。硬件环境是数据库系统的物理支撑。数据库系统需要在操作系统的支持下工作，其本身也包含着数据库管理系统和应用程序等，需要有较大的内存容量。同时，由于用户的数据、系统软件和应用软件都要保存在外部存储器上，因此也需要较大的外存容量。

(2) 软件系统。软件系统包括系统软件和应用软件。系统软件主要包括支持数据库管



理系统运行的操作系统、数据库管理系统、开发应用系统的程序设计语言及其编译系统、应用系统开发工具等。系统软件为开发应用系统提供了良好的环境。应用软件是指在数据库管理系统的基础上,由用户根据自己的实际需要自行开发的应用程序。

(3) 数据。数据是数据库的基本组成内容,是数据库系统的管理对象,是用户的信息源。数据库是把数据按一定的结构和形式组织起来的记录的集合。由于数据类型的多样性,数据的采集方式和存储方式也会有所不同,数据作为一种资源是数据库系统最稳定的成分。

(4) 人员。数据库系统的人员是指参与分析、设计、管理、开发、维护和使用数据库的主要人员,也是数据库系统的组成部分,主要包括数据库管理员、系统分析员、应用程序员和用户。数据库管理员负责管理和控制数据库系统;系统分析员负责应用系统的需求分析和规范说明,确定系统的软硬件配置、系统的功能及数据库概念设计;应用程序员负责设计应用系统的程序模块,编写应用程序;最终用户通过系统提供的用户接口界面使用数据库,常用的接口方式有菜单驱动、图形显示和表格操作等,这些接口为用户提供了简明直观的数据表示方式和方便快捷的操作方法。

### 2.1.3 数据模型

现实世界中的客观事物是相互联系的。一方面,事物内部的因素和属性根据一定的组织原则相互联系,构成一个相对独立的系统;另一方面,事物本身也作为更大系统的一个因素或一种属性而存在,并与系统的其他因素或属性相联系。客观事物的这种普遍联系性,决定了作为事物属性和记录符号的数据与数据之间也存在着一定的联系。具有联系的数据总是按照一定的组织关系排列,从而构成一定的结构,这种结构的描述就是数据模型。

数据模型规范了数据库中数据的组织形式,并展示了数据与数据之间的关系。数据模型的好坏直接影响数据库的性能。支持数据库系统的常用数据模型有层次模型(hierarchical model)、网状模型(network model)、关系模型(relational model)及面向对象模型(object oriented model)。

#### 1. 层次模型

现实世界中很多事物是按层次组织起来的。层次数据模型的提出,首先是为了模拟这些按层次组织起来的事物。层次数据库也是按记录来存取数据的。层次数据模型中最基本的数据关系是基本层次关系,它代表两个记录之间一对多的关系,在层次模型中,每个数据元素可以与下面任何一层的多个数据元素相联系,但只能与它上面一层中的一个数据元素相联系。层次模型最简单的结构及约束,可以用数据结构中的树来描述。层次模型像一棵由根元素出发向下逐层辐射枝叶的倒置的树,最高一层的数据元素称为根元素,根元素是唯一的,而且只有下属元素。树的节点表示数据元素,枝表示数据元素之间的联系。在层次模型中,有且仅有一个节点无上层节点(称为根),其他节点有且仅有一个上层节点。层次模型结构如图 2-1 所示。

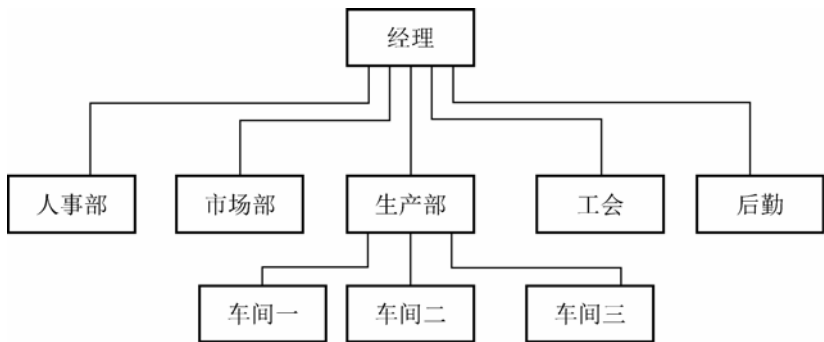


图 2-1 层次模型结构

## 2. 网状模型

在网状模型中，可以有一个以上的数据元素没有上层联系，至少有一个以上的节点有多于一个的上层节点。因此，网状模型更具有表示事实的能力，可以用来描述事物之间较复杂的联系。网状模型适用于表达事物之间的“多对多”型的联系。由各供应商与公司构成的一个网状模型的例子，如图 2-2 所示。

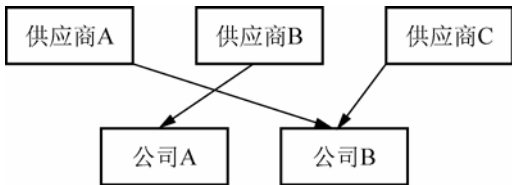


图 2-2 网状模型示意图

## 3. 关系模型

在关系模型中，不同数据之间的联系用关系来表示，其实质是将数据的逻辑结构归结为满足一定条件的二维表形式，每个二维表称为一个关系。在二维表中，每行称为一个记录（元组），每列称为一个数据项或字段（属性），数据项名称为字段名或属性名，整个二维表表示一个关系。这种模型既可以用来表达事物之间“一对一”或“一对多”的联系，也可以用来表达“多对多”的联系。

关系模型用二维表的逻辑结构来表达数据及数据之间的联系。例如，有一组数据如下。

学号，JX0601，王磊，男，20 岁，VFP，79，平面设计，89，网页制作，90，平均分，86

学号，JW0602，张芳芳，女，18 岁，VFP，83，平面设计，88，网页制作，92，平均分，87

学号，WH0603，马晓倩，女，19 岁，VFP，80，平面设计，81，网页制作，83，平均分，81

这 3 组数据之间是平行的，互不从属，但假如我们知道这些数据来自于同一个班级的学生，便可以构成某学校某班学生基本情况表，并建立一个关系。下面以表 2-1 所示的某班学生成绩表为例来分析关系数据模型。



表 2-1 某班学生成绩表

| 学 号    | 姓 名 | 性 别 | VFP | 平 面 设 计 | 网 页 制 作 | 总 分 | 平 均 分 |
|--------|-----|-----|-----|---------|---------|-----|-------|
| JX0601 | 王 磊 | 男   | 79  | 89      | 90      | 258 | 86    |
| RJ0602 | 张芳芳 | 女   | 83  | 88      | 92      | 261 | 87    |
| WH0603 | 马晓倩 | 女   | 80  | 81      | 83      | 243 | 81    |
| WL0604 | 李小莉 | 女   | 67  | 78      | 80      | 225 | 75    |
| WL0605 | 张国明 | 男   | 85  | 88      | 91      | 264 | 88    |
| WL0606 | 付会军 | 男   | 78  | 80      | 89      | 247 | 82    |
| RJ0607 | 贺玉红 | 女   | 67  | 60      | 70      | 197 | 67    |

表 2-1 中的数据虽然是平行的,不具有从属关系,但却构成了某个班级的学生之间的关系结构。

同样地,假设上述 7 组数据不是来自同一个班级,而是来自不同的专业系,便可以构成某学院各专业学生基本情况表,又可以建立一个关系,如表 2-2 所示。

表 2-2 某学院各专业学生基本情况表

| 学 号    | 姓 名 | 性 别 | 年 龄 | 专 业 名 称 |
|--------|-----|-----|-----|---------|
| JX0601 | 王 磊 | 男   | 20  | 信息管理    |
| RJ0602 | 张芳芳 | 女   | 18  | 软件工程    |
| WH0603 | 马晓倩 | 女   | 19  | 应用与维护   |
| WL0604 | 李小莉 | 女   | 20  | 网件技术    |
| WL0605 | 张国明 | 男   | 20  | 网件技术    |
| WL0606 | 付会军 | 男   | 19  | 网件技术    |
| RJ0607 | 贺玉红 | 女   | 20  | 软件工程    |

以上两表所表现的都是关系模型。表格中的每列数据都是独立的数据项,它们共同构成了该关系的全部内容。

表格中的一行称为一个记录(record)。记录用来表示关系模型中若干平行的、相对独立的个体事物,每个记录由若干数据项的值组成。

表格中的一列称为一个字段(field),用来描述某一个体某方面的属性,它是表示关系模型中具有某种属性的一列数据。通常表格的第一行表示属性类型的名称,称为字段名。

总体来讲,以竖向的数据属性分类的若干记录的集合,构成一个关系模型,称为一个关系(relation)。在某种意义上,关系模型就是一张二维表,用来描述客观事物属性的关系。我们通过分析得出关系模型主要具有以下特点。

(1) 一个关系就是一张二维表,在关系中每个数据项都不可再分,数据项是最基本的单位。

(2) 同一列数据项具有相同的类型,这被称为数据的属性。各列都有唯一的属性名和对应的属性值。

(3) 行数据是个体事物诸多属性值的集合,被称为记录。

(4) 行和列的顺序可以是任意的。

(5) 在一个关系中,不允许有相同的字段名,也不允许有完全相同的记录行。

通常将一张二维表称为一个数据表,数据表中包含了数据和数据之间的关系。

层次模型和网状模型是早期的数学模型,已被逐渐淘汰。由于关系模型更为简单灵活,因此目前流行的数据库软件大多使用关系模型。但是,随着信息的大量传播,现实生活中

存在着许多更复杂的数据结构和应用领域，使用关系模型处理这些复杂的数据较为困难，因此产生了面向对象数据模型。面向对象数据模型是正在发展中的具有广泛应用开发价值的模型，但目前在此基础上建立的数据库大都是实验性的。

#### 2.1.4 关系数据库

##### 1. 关系数据库概述

一个关系数据库由若干数据表组成，一个数据表由若干记录构成，而一条记录又由若干字段（数据项）组成。在关系数据库中，每个数据表都具有相对独立性，该独立性的唯一标志是数据表名，又称表文件名。一个表文件名代表一个独立的表文件。因为在关系数据库中，对数据表中数据的访问首先通过表文件名实现，所以不允许数据表同名。关系数据库中各个数据表都具有独立性，操作人员可以方便地对数据表中的数据进行编辑、存取和传输。

关系数据库中各数据表之间的相关性，是依靠各个独立数据表内具有相同属性的字段而建立起来的。一般情况下，一个关系数据库中可以有多个相关的独立数据表，这为数据资源的共享和充分利用提供了极大的方便。关系数据库中各数据表之间具有独立性，而若干个数据表之间又具有相关性，利用关系模型构筑的关系型数据库，可以提供数据库系统的全部功能，关系数据库由于这一极大的优越性而得到迅速普及。

关系数据库（Relation database）是若干按照关系模型设计的数据表文件的集合，也可以说，关系数据库是由若干按关系模型设计的二维表组成的。关系型数据库除了功能强大以外，还具有操作简单、易学易懂和数据独立性强等优点，近年来得到了迅速的发展和推广普及。

##### 2. 关系模型中的几个基本概念

关系数据库采用关系模型作为数据的组织方式。关系数据库的基本数据结构是满足一定条件的二维表。

（1）属性（Attribute）：二维表中的一列称为一个属性，又称字段（Field）。

（2）元组（Tuple）：二维表中的一行称为一个元组，又称记录（Record）。

（3）关键字：属性及属性的组合，即二维表中的某个属性。其值能唯一地标志一个元组，表示为字段与字段的组合，假定一个企业中的职工号无重复，则职工号就是人事关系中唯一能标志一个元组的关键字。若一个关系中有多个关键字都能唯一标志一个元组，可选定其中一个为“主关键字”，这个属性称为主属性，其余的为候选关键字，主关键字和候选关键字都能起到唯一标志元组的作用。

（4）关系（Relation）：一个二维表称为一个关系，元组的集合也称为关系。

（5）域（Domain）：一个属性的取值范围称为一个域。

（6）分量：元组中的一个属性值称为元组的一个分量。

（7）关系模式：对关系的描述，包括关系名、组成该关系的属性名和属性到域的映像。通常简记为：关系名（属性名<sub>1</sub>，属性名<sub>2</sub>，…，属性名<sub>n</sub>）。

如前所述，关系数据库是采用关系模式作为数据组织方式的数据库。对关系数据库的描述，称为关系数据库模型，它包括若干域的定义及在这些域上定义的若干关系模式。这些关系模式在某一时刻对应的关系的集合，称为关系数据库的值。

例如，表 2-1 中的关系是一个学生成绩表。表中的一行是关系的一个元组，“学号”、“姓





名”、性别、“平面设计”、“网页制作”等是属性。其中，“学号”是识别一条记录的主属性，称为“主关键字”。在表 2-1 中，“学号”属性的域是 0601~0607；“姓名”属性的域是 2~4 个汉字组成的字符串；“性别”属性的域是“男”或“女”；“平均分”属性的域是 0~100 等。学生成绩表的关系模式可表示为：

学生成绩表（学号，姓名，性别，VFP，平面设计，网页制作，总分，平均分）

### 3. 关系运算

关系之间可以进行运算，关系运算可以用两种方式表示：关系代数和关系演算。现已证明这两种方式是等价的，这里仅以关系代数来表示关系运算。

关系代数是把关系作为“集合”来进行运算的。常用的关系运算有两类：传统的集合运算和专门的关系运算。

（1）传统的集合运算有集合的并、交、差等，其运算是从关系二维表的行的角度来进行的。

集合的并运算。两个关系集合的并运算，是将这两个关系合并，并消去冗余元组。例如，表 2-3 和表 2-4 所示的是两个特殊的关系 A 和 B（学生成绩处理系统中部分有关的关系），给这两个关系做集合的并运算时，要求这两个关系的所有属性相同。并运算的结果是将这两个关系合并，并消去冗余元组，运算结果如表 2-5 所示。

表 2-3 学生学籍关系 A

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| JX0601 | 王 磊 | 86    |
| RJ0602 | 张芳芳 | 87    |
| WH0603 | 马晓倩 | 81    |
| WL0604 | 李小莉 | 75    |

表 2-4 学生学籍关系 B

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| WL0604 | 李小莉 | 75    |
| WL0605 | 张国明 | 88    |
| WL0606 | 付会军 | 82    |
| RJ0607 | 贺玉红 | 67    |
| JX0601 | 王 磊 | 86    |

表 2-5 A 与 B 并运算的结果

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| JX0601 | 王 磊 | 86    |
| RJ0602 | 张芳芳 | 87    |
| WH0603 | 马晓倩 | 81    |
| WL0604 | 李小莉 | 75    |
| WL0605 | 张国明 | 88    |
| WL0606 | 付会军 | 82    |
| RJ0607 | 贺玉红 | 67    |

集合的交运算。两个关系做集合的交运算时，同样要求关系中的所有属性相同。给两个关系做交运算，就是取两个关系中的公共元组。A 与 B 交运算的结果，如表 2-6 所示。

表 2-6 A 与 B 交运算的结果

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| JX0601 | 王 磊 | 86    |
| WL0604 | 李小莉 | 75    |

集合的差运算。两个关系做集合的差运算，也要求关系中的所有属性相同。两个关系差运算的结果是属于关系 A 但不属于关系 B 的元组，A 与 B 差运算的结果，如表 2-7 所示。

表 2-7 A 与 B 差运算的结果

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| RJ0602 | 张芳芳 | 87    |
| WH0603 | 马晓倩 | 81    |

(2) 专门的关系运算有选择、投影和连接等，这类运算不仅涉及关系的水平方向（二维表的行），而且涉及关系的竖直方向（二维表的列）。

选择运算（Selection）。选择运算是在关系中选择某些条件的元组，即选择满足条件的行。例如，在学生成绩表中选取所有男生的元组，按二维表的行进行，选择的条件是“性别”等于“男”，可用选择运算来实现，选择运算的结果如表 2-8 所示。选择运算常用于对指定关系按条件进行检索。

表 2-8 选择运算的结果

| 学 号    | 姓 名 | 性 别 | 年 龄 | 专 业 名 称 |
|--------|-----|-----|-----|---------|
| JX0601 | 王 磊 | 男   | 20  | 信息管理    |
| WL0605 | 张国明 | 男   | 20  | 网件技术    |
| WL0606 | 付会军 | 男   | 19  | 网件技术    |

投影运算（Projection）。投影运算是在关系中选择某些属性的列，其余属性的列都不在运算结果中。例如，在表 2-1 所示的某班学生成绩表中找出平均分在 85 分以上的所有学生的学号、姓名和平均分，就可以用投影运算实现，投影运算的结果如表 2-9 所示。投影运算用来检索指定关系中用户所关心的属性所在列的值，运算结果是一个新的关系。

表 2-9 投影运算的结果

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| JX0601 | 王 磊 | 86    |
| RJ0602 | 张芳芳 | 87    |
| WL0605 | 张国明 | 88    |

连接运算（Join）。连接是关系的横向组合，连接运算是将两个关系中的属性拼接成一个更宽的关系模式，生成的新关系中包含满足条件的元组。连接过程通过连接条件来控制，连接条件中将出现两个关系的公共属性，连接选取属性间满足一定条件的元组。例如，在表 2-10（关系 E）和表 2-11（关系 F）中选取满足条件的元组构成表 2-12（关系 G）。



表 2-10 关系 E

| 学 号    | 姓 名 | 平 均 分 |
|--------|-----|-------|
| JX0601 | 王 磊 | 86    |
| RJ0602 | 张芳芳 | 87    |
| WH0603 | 马晓倩 | 81    |
| WL0604 | 李小莉 | 75    |

表 2-11 关系 F

| 学 号    | 性 别 | 专 业 名 称 |
|--------|-----|---------|
| JX0601 | 男   | 信息管理    |
| RJ0602 | 女   | 软件工程    |
| WH0603 | 女   | 应用与维护   |
| WL0604 | 女   | 网件技术    |

表 2-12 关系 G

| 学 号    | 姓 名 | 性 别 | 专 业 名 称 | 平 均 分 |
|--------|-----|-----|---------|-------|
| JX0601 | 王 磊 | 男   | 信息管理    | 86    |
| RJ0602 | 张芳芳 | 女   | 软件工程    | 87    |
| WH0603 | 马晓倩 | 女   | 应用与维护   | 81    |
| WL0604 | 李小莉 | 女   | 网件技术    | 75    |

对关系 E 和关系 F 进行条件为“学籍号 = 学籍号”的连接操作，所得关系 G 是关系 F 的一个子集。

连接条件中的属性称为连接属性，两个关系中的连接属性应有相同的数据类型，以保证其可比性。若连接条件中的运算表达式为“学籍号 = 学籍号”，则为等值连接。表 2-12 是关系 E 和关系 F 在条件“学籍号 = 学籍号”下的等值连接。若在等值连接的结果中去掉重复的属性（或属性组），则称为自然连接。表 2-12 所示的关系 G 也是在条件“学籍号 = 学籍号”下的自然连接。

由此可见，关系操作的特点是集合操作，即操作的对象是元组的集合关系，关系代数的运算对象是关系，操作的结果仍然是关系。利用关系代数的运算，可以产生所需要的新关系，或进行指定条件的检索（选择、投影）。对关系数据库的实际操作，常常是以上几种运算的综合应用。

## 任务 2.2 实例训练

### 2.2.1 创建“学籍管理数据库”

Visual FoxPro 使用数据库管理表。创建数据库有 3 种方法：使用项目管理器，使用数据库向导，使用命令。

#### 1. 在项目管理器中创建数据库

操作步骤如下。

(1) 选择“文件”|“打开”命令。

(2) 弹出“打开”对话框，选择项目管理器名称，这里选择“学生学籍管理系统.pjx”，单击“确定”按钮。

(3) 弹出如图 2-3 所示的项目管理器，选择“数据库”选项。

(4) 单击“新建”按钮后，弹出如图 2-4 所示的“新建数据库”对话框，单击“新建数据库”按钮。



图 2-3 项目管理器



图 2-4 “新建数据库”对话框

(5) 弹出“创建”对话框。在该对话框的“保存类型”下拉列表框中，选择文件类型“数据库(\*.dbc)”，在“数据库名”文本框中，输入新数据库的名称，这里输入“学籍管理数据库”，并单击“保存”按钮。

(6) 此时弹出如图 2-5 所示的“数据库设计器”窗口。



图 2-5 “数据库设计器”窗口

(7) 该窗口中有一个“数据库设计器”工具栏，用户可利用该工具栏建立各种类型的数据文件，如表、视图、连接和存储过程，并将它们添加到“学籍管理数据库”中。这里只是创建了一个空的数据库。

(8) 单击数据库设计器右上角的关闭按钮将其关闭。这样，“学籍管理数据库”这个数据库文件就在项目管理器中创建了。

## 2. 使用菜单创建数据库

操作步骤如下。

(1) 单击“常用”工具栏上的“新建”按钮或选择“文件”|“新建”命令，打开“新建”对话框，如图 2-6 所示。



(2) 选中“数据库”单选按钮,然后单击“新建”按钮,打开“创建”对话框,如图 2-7 所示。

(3) 设置好保存文件的文件夹和数据库名之后,单击“保存”按钮即可。



图 2-6 “新建”对话框

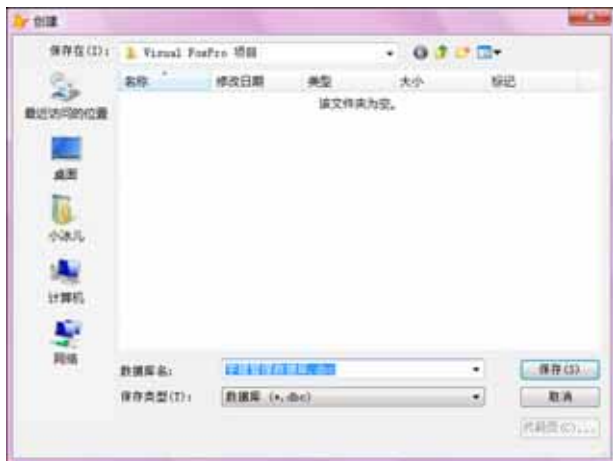


图 2-7 “创建”对话框

### 3. 使用命令创建数据库

使用菜单创建数据库时,系统自动在命令窗口中生成创建数据库的 CREATE DATABASE 命令。创建数据库命令的基本格式如下。

CREATE DATABASE <数据库名>

命令功能: 创建一个数据库。

参数说明: <数据库名>是指定要创建的数据库文件名,其中可以包含路径。

例如,要在 F:\学生学籍管理系统文件夹中创建一个文件名为“学籍管理数据库.dbc”的数据库,可以执行以下命令。

CREATE DATABASE F:\学生学籍管理系统\学籍管理数据库.dbc

使用命令创建数据库时, Visual FoxPro 并不打开数据库设计器。

#### 2.2.2 打开“学籍管理数据库”

使用数据库之前要先打开数据库。Visual FoxPro 可以打开多个数据库。用户可以在项目管理器中打开数据库,也可以使用菜单和命令打开数据库。

(1) 使用项目管理器打开数据库。在项目管理器中,单击“数据库”前面的“+”号,展开其选项,然后单击“学籍管理数据库”前面的“+”号,展开其选项,可以打开该数据库。先选择“学籍管理数据库”,再单击“打开”按钮,也可以打开该数据库。使用项目管理器打开数据库如图 2-8 所示。

(2) 使用菜单打开数据库。单击“常用”工具栏上的“打开”按钮或选择“文件”|“打开”命令,可以使用打开文件的一般方法打开数据库。使用这种方法打开数据库时,数据库设计器将自动打开。

(3) 使用命令打开数据库。使用菜单打开数据库文件时,系统自动在命令窗口生成打开数据库的 OPEN DATABASE 命令,其基本格式如下。



图 2-8 使用项目管理器打开数据库

OPEN DATABASE <数据库名>

命令功能: 打开指定的数据库。

例如,要打开默认目录下的“学籍管理数据库”这个数据库文件,可以执行以下命令。

OPEN DATABASE 学籍管理数据库.dbc

使用命令打开数据库时, Visual FoxPro 并不自动打开数据库设计器。

### 2.2.3 设置“学籍管理数据库”为当前数据库

Visual FoxPro 虽然允许同时打开多个数据库,但在任何时刻,用户都只能对其中的一个数据库进行操作,该数据库称为当前数据库。系统默认最后打开的数据库为当前数据库。当要使用的数据库不是当前数据库时,可以使用下列方法将其设置为当前数据库。

(1) 使用工具栏设置当前数据库。单击“常用”工具栏中的“数据库”下拉列表框的下拉按钮,从该下拉列表选择一个数据库,即可将其设置为当前数据库。

(2) 使用命令设置当前数据库。设置当前数据库命令的基本格式如下。

SET DATABASE TO <数据库名>

命令功能: 设置当前数据库。

参数说明: 选择<数据库名>,指定该数据库为当前数据库。默认该选项,则关闭当前数据库。

例如,设置“学籍管理数据库”为当前数据库,可以执行以下命令。

SET DATABASE TO 学籍管理数据库.dbc

### 2.2.4 关闭“学籍管理数据库”

在 Visual FoxPro 中,用户使用完数据库后,要将其关闭。使用下列方法可以关闭数据库。

(1) 在项目管理器中关闭数据库。先选择要关闭的数据库“学籍管理数据库”,再单击“关闭”按钮。在项目管理器中关闭数据库如图 2-9 所示。

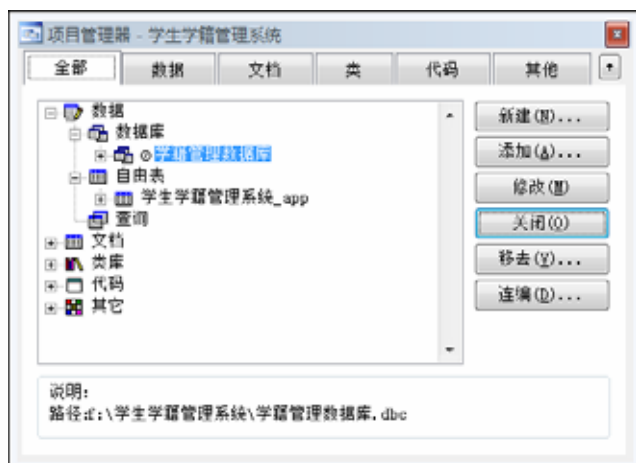


图 2-9 在项目管理器中关闭数据库

(2) 使用命令关闭一个数据库。关闭当前数据库的命令的基本格式如下。

**CLOSE DATABASE**

命令功能：关闭当前数据库。

CLOSE DATABASE 命令只能关闭当前数据库。如果要关闭的数据库不是当前数据库，可以先将其设置为当前数据库再关闭。

(3) 使用命令关闭多个数据库。关闭多个数据库命令的基本格式如下。

**CLOSE ALL**

命令功能：关闭所有打开的文件。

### 2.2.5 向“学籍管理数据库”中添加表

有了数据库文件，就可以向数据库添加自由表，使之转为数据库表。

#### 1. 操作步骤

(1) 打开“学籍管理数据库”，在项目管理器中双击“学籍管理数据库”，或者选择“学籍管理数据库”，单击“修改”按钮，进入“数据库设计器”窗口，如图 2-10 所示。



图 2-10 “数据库设计器”窗口

(2) 单击数据库设计器中的“添加表”按钮，或者单击鼠标右键弹出数据库快捷菜单，选择“添加表”选项，弹出“选择表名”对话框。

(3) 在“选择表名”对话框中,选择要添加的表,再单击“确定”按钮,返回“数据库设计器”窗口。

## 2. 实例说明

例如,将学生信息表 student 添加到已建立的“学籍管理数据库”中,操作步骤如下。

(1) 在项目管理器中选择数据库“学籍管理数据库”,单击“修改”按钮,打开“数据库设计器”窗口。

(2) 在“数据库设计器”窗口中单击鼠标右键,弹出“数据库设计器”快捷菜单窗口,如图 2-11 所示,选择“添加表”选项,进入“选择表名”对话框。



图 2-11 “数据库设计器”快捷菜单窗口

(3) 在“选择表名”对话框,选择要添加的学生信息表 students,再单击“确定”按钮,返回“数据库设计器”窗口,这时“学籍管理数据库”中就有了一个表,添加一个表后的“学籍管理数据库”如图 2-12 所示。

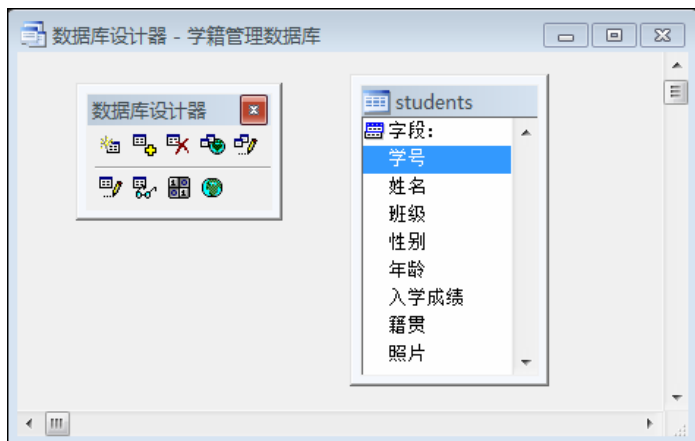


图 2-12 添加一个表后的“学籍管理数据库”





### 2.2.6 修改“学籍管理数据库”

操作步骤如下。

在项目管理器中，选择数据库“学籍管理数据库”，单击“修改”按钮，打开“数据库设计器”窗口，即可在其中修改数据库。

### 2.2.7 删除“学籍管理数据库”

操作步骤如下。

在项目管理器中，选择数据库“学籍管理数据库”，单击“移去”按钮，弹出如图 2-13 所示的移去数据库提示对话框。如果单击“移去”按钮，那么数据库从该项目中移去；如果单击“删除”按钮，那么数据库从磁盘中删除。



图 2-13 移去数据库提示对话框

另外，还可以使用命令删除数据库。删除数据库命令的基本格式如下。

DELETE DATABASE DatabaseName | ? [DELETETABLES] [RECYCLE]

命令功能：从磁盘中删除一个数据库。

参数说明如下。

(1) DatabaseName 指定要从磁盘中删除的数据库名。指定的数据库不能处于打开状态。DatabaseName 可以包含指定数据库的路径和数据库名。

(2) “?” 为显示删除对话框，可以从该对话框中指定要从磁盘中删除的数据库名。

(3) DELETETABLES 指定从磁盘中删除的包含在数据库中的表和包含表的数据库。

(4) RECYCLE 指定数据库不是直接从磁盘中删除，而是放入 Windows 回收站。

## 任务 2.3 项目分析

数据库在 Visual FoxPro 中占有很重要的位置，它将各种数据表组织在一起进行管理，并使之相互关联。在本项目中，我们在“学生学籍管理系统”的基础上，建立了“学籍管理数据库”，至此，学生学籍管理系统的目录结构如图 2-14 所示。



图 2-14 学生学籍管理系统的目录结构

## 任务 2.4 模仿设计与自主设计

### 2.4.1 数据库的创建

(1) 在图书管理系统中，创建一个“图书管理”数据库。分别使用以下 3 种方法。

在项目管理器中创建“图书管理”数据库。

使用菜单创建“图书管理”数据库。

使用命令创建“图书管理”数据库。

(2) 打开“图书管理”数据库，并设置“图书管理”数据库为当前数据库。

### 2.4.2 数据库中表的操作

(1) 在“自由表”选项下建立以下表：图书信息表、借阅者信息表和借阅者身份表。

(2) 向“图书管理”数据库中添加上述自由表。

(3) 移去图书信息表，删除借阅者信息表和借阅者身份表。

## 项目小结 2

项目 2 主要介绍了数据库的基础知识，包括数据、信息、数据管理、数据模型和与数据库系统相关的一些基础理论知识，特别介绍了关系数据模型的相关知识，并较为完整地介绍了在 Visual FoxPro 中建立数据库、修改数据库及删除数据库的方法。这为后面学习和掌握 Visual FoxPro 9.0 程序设计技术打下了基础。



## 项目习题 2

### 1. 选择题

(1) Visual FoxPro DBMS 是 ( )。

- A. 操作系统的一部分      B. 一种编译程序  
C. 操作系统支持下的系统软件      D. 一种操作系统

(2) 设有部门和职员两个实体, 每个职员只能属于一个部门, 一个部门可以有多名职员, 则部门与职员之间的联系类型是 ( )。

- A. m:n      B. 1:m      C. m:k      D. 1:1

(3) 对于“关系”的描述, 正确的是 ( )。

- A. 同一个关系中允许有完全相同的元组  
B. 同一个关系中元组必须按关键字升序存放  
C. 在一个关系中必须将关键字作为该关系的第一个属性  
D. 同一个关系中不能出现相同的属性名

(4) 关系数据库管理系统所管理的关系是 ( )。

- A. 一个 DBF 文件      B. 若干二维表  
C. 一个 DBC 文件      D. 若干 DBC 文件

(5) 关系的 3 种基本运算方法指 ( )。

- A. 选择、投影、连接      B. 选择、投影、添加  
C. 选择、连接、删除      D. 连接、投影、插入

(6) 在 Visual FoxPro 中, 数据库文件的扩展名为 ( )。

- A. .dbc      B. .dct      C. .dcx      D. .dbf

(7) 下列创建数据库的方法, 正确的是 ( )。

- A. 在“项目管理器”中选定“数据”选项卡, 选择“数据库”选项, 单击“新建”按钮  
B. 在“新建”对话框中选中“数据库”单选按钮, 单击“新建文件”按钮  
C. 在命令窗口中输入 CREATE DATABASE<数据库文件名>  
D. 以上方法都可以

(8) 在 Visual FoxPro 中, 打开一个数据库文件的命令是 ( )。

- A. CREATE DATABASE <数据库名>  
B. OPEN DATABASE <数据库名>  
C. CREATE <数据库名>  
D. OPEN <数据库名>

(9) 在 Visual FoxPro 中, 以只读方式打开数据库文件的命令是 ( )。

- A. EXCLUSIVE      B. SHARED  
C. NOUPDATE      D. VALIDATE

- (10) 在 Visual FoxPro 中, 打开数据库设计器的命令是 ( )。
- A . OPEN DATABASE                      B . USE DATABASE  
C . CREATE DATABASE                      D . MODIFY DATABASE

## 2. 填空题

- (1) 用树形结构表示实体之间联系的模型是\_\_\_\_\_。
- (2) 数据模型不仅表示反映事物本身的数据, 而且表示\_\_\_\_\_。
- (3) Visual FoxPro 不允许在主关键字中有重复值或\_\_\_\_\_。
- (4) 数据库系统的核心是\_\_\_\_\_。
- (5) 自然连接是指\_\_\_\_\_。



# 项目 3

## 设计学生学籍管理 系统中的表

### 项目要点

- 数据表的有关概念。
- 数据表的基本操作。

### 学习目标

- 了解数据的完整性等概念。
- 掌握表的建立和基本操作。
- 掌握索引的建立和使用方法。
- 掌握表的排序方法。

在关系数据库管理系统中，数据库的操作是在表的基础上进行的。表结构的设计直接影响表的使用效果，数据表中数据的冗余度、共享性和完整性直接影响表的质量。

Visual FoxPro 有两种类型的表：数据库表和自由表。数据库表是某个数据库的一部分，而自由表则是一个独立的表，不属于任何数据库。与自由表相比，数据库表有一些特殊的性质。项目 2 中已经介绍了数据库中的表可以从数据库中移去，变成自由表；自由表可以添加到某个数据库中，成为一个数据库表。本项目以创建学生学籍管理系统的表为例，介绍如何创建表，以及表的基本操作。

## 任务 3.1 熟悉表的基础知识

### 3.1.1 表的结构

在 Visual FoxPro 系统中，一张二维表对应一个数据表文件（table）。一个表由表名、表的字段和表的记录 3 个要素构成。

#### 1. 定义表名

定义表名，就是定义表文件的名称，它是该文件存储到磁盘的唯一标志。表名要能体现出文件的内容，也要便于使用。

#### 2. 定义表的字段属性

表的字段属性即表的组织形式，包括表的字段个数和每个字段的名称、类型、宽度、小数位数及是否建立索引等。

#### 3. 向表中输入数据

表中的数据是表的记录，记录的内容是表提供给用户的全部信息。向表中输入数据就是为表中记录的每个字段赋值。不包含记录的表为空表。

创建一个新表时，最关键的一步是确定表中的字段，即定义表的结构。在下一个任务中，我们将以学生学籍管理系统中的课程表为例，向大家介绍如何在 Visual FoxPro 中建立表并进行表的各项操作。

### 3.1.2 自由表和数据库表

#### 1. 自由表

所谓自由表结构，是指用户在某个项目管理器中的“数据”选项卡的“自由表”选项下创建的表结构。之所以称它为自由表，是因为它不与其他表发生联系，既不能控制其他表，也不被其他表所控制，是独立存在于任何数据库之外的表。

利用表设计器可以方便地创建表的结构。

#### 2. 数据库表

所谓数据库表结构，是指用户在某个项目管理器中的某个数据库下创建的表结构。当一个表是数据库的一部分时，它就可以具有：

- (1) 长表名和表中的长字段名；
- (2) 表中字段的标题和注释；
- (3) 默认值、输入掩码和表中字段格式化；
- (4) 表字段的默认控件类；
- (5) 字段级规则和记录级规则；
- (6) 支持参照完整性的主关键字索引和表间关系；
- (7) INSERT、UPDATE 或 DELETE 事件的触发器。



### 3.1.3 修改表结构

在操作使用表的过程中，常常会因为创建表时考虑不周或操作不慎，导致表结构不能适应新情况，此时可以修改表结构。

利用表设计器可以改变已有数据表的结构，如增加或删除字段，改变字段的数据类型和宽度，查看表的内容和设置索引等。如果修改的数据表是数据库的一部分，还可以得到附加的与数据库有关的字段和数据表的属性。

修改表结构主要包括以下几个方面的内容。

- (1) 修改表中字段的属性。
- (2) 在表中增加字段。
- (3) 删除表中字段。
- (4) 改变字段顺序。

### 3.1.4 索引

若按特定顺序定位、查看或操作表中的记录，可以使用索引。Visual FoxPro 将索引作为排序机制，为开发应用程序提供灵活性。根据应用程序的要求，用户可以灵活地对同一个表创建和使用不同的索引关键字，这样就可以按不同的顺序处理记录，也可以根据这些索引自定义表间的关系，准确地访问想要的记录。

#### 1. 索引的概念

索引是用户根据数据库表中的某些字段值，为数据表建立一个顺序，使记录的显示、查询和打印更为迅速，同时，也可以限制记录重复，并支持不同表之间的“关系”。由于索引文件中存储的是按照某一字段的值排列的一组记录号，每个记录号指向一个待处理的记录，因此，索引可以理解为根据某一字段的值进行逻辑排序的一组指针。它与通常的图书索引目录相同，图书中的索引指明了章、节的页码，而表的索引指明了由某一字段值的大小决定记录的排列顺序。

索引文件和表的.dbf 文件分别存储，并且不改变表中记录的物理顺序。实际上，创建索引就是创建一个由指向.dbf 文件记录的指针构成的文件。若要根据特定顺序处理表记录，可以选择一个相应的索引。

#### 2. 索引关键字

关键字是用来作为索引顺序的字段名。例如，对学生信息表中的“学号”进行索引，建立一个按“学号”大小排序的索引，那么“学号”就是索引的关键字。

#### 3. 索引关键字的类型

- (1) 在 Visual FoxPro 9.0 中，索引关键字可分为以下几种类型。

主索引(primary index)。主索引是指关键字段或索引表达式中不允许出现重复值的索引，主要用于主表或被引用的表，用来在一个永久关系中建立参照完整性。一个表只能创建一个主索引，并且主索引是对于数据库表而言的，自由表只有后面 3 种索引类型。

例如，学号、课程号都可以作为主索引字段，但其他的字段如姓名、性别、班级和开设学期等都不可以作为主索引，因为它们的输入值并不是唯一的。

候选索引 (candidate index)。候选索引是指可以作为主关键字的索引,因为它不包含 NULL 值或重复值。在数据库表和自由表中均可以为每个表建立多个候选索引。

唯一索引 (unique index)。唯一索引是指不允许两个索引具有相同索引值的索引,这种要求与主索引相同。为了保持与早期版本的兼容性,可以建立一个唯一索引,以指定字段的首次出现值为基础,选定一组记录,并对记录进行排序。

普通索引 (regular index)。普通索引是指可以用来对记录排序和搜索记录的索引,它不强迫记录中的数据具有唯一性,一个表中可以有多个普通索引。

例如,在学生成绩表的“学号”、“课程号”和“成绩”3个字段中,“学号”和“课程号”可以定义为普通索引。

从以上的定义可以看出,主索引和候选索引具有相同的功能,它们除了具有按升序和降序索引的功能外,都具有关键字的特性,建立主索引和候选索引的字段值可以保证其唯一性。此外,要特别注意唯一索引的含义,唯一索引与字段值的唯一性无关,即建立了唯一索引的字段,它的字段值是可以重复的,它的“唯一”是指在使用相应的索引时,重复的字段值只有唯一的一个出现在索引项中。

(2) 选择合适的索引类型可以将下列准则作为依据。

如果需要排序记录,以便显示、查询或打印,那么可以使用普通索引、候选索引或主索引。

如果要在字段中控制重复值的输入并对记录排序,那么可以对数据表使用主索引或候选索引,对自由表使用候选索引。

如果准备设置关系,那么可以依据表在关系中所起的作用分别使用普通索引、主索引或候选索引。

#### 4. 索引文件的分类

(1) 独立索引文件。独立索引文件 (.idx) 只有一个索引表达式。用户在使用这种类型的索引文件时,必须先打开索引文件,否则,数据表中数据的变化不会引起索引文件的相应变化。独立索引文件是为了与低版本的数据库管理系统兼容而提供的索引文件类型。

(2) 复合索引文件。复合索引文件可以在同一个数据表中创建多个索引关键字,从而在一个索引文件中存放多个索引,以满足一个数据表可能有多种索引顺序的需求。每个索引关键字表达式对应一个索引标志符 (index tag), 每个索引标志符的效果等同于一个索引文件,可以排列数据或查找记录。数据表的变动引起相关索引顺序变动时,索引标志符也自动调整,这相当于将若干索引文件复合成一个索引文件,故称为复合索引。复合索引文件对其中可存放的索引标志符的个数并无限制,其个数仅受内存和磁盘空间的限制,且每个索引标志符均是自动压缩的。

复合索引文件分为两种:一是结构复合索引文件,二是非结构复合索引文件。

结构复合索引文件 (structure compound index)。

结构复合索引文件的主文件名与数据表的主文件名相同,但扩展名为.cdx。结构复合索引文件在打开数据表时自动打开,且能自动调整。

非结构复合索引文件 (independent compound index)。

非结构复合索引文件的主文件名与数据表的主文件名不相同。非结构复合索引文件不能随着数据表的打开而自动打开,必须用命令打开,其余部分与结构复合索引文件没有区别。





### 3.1.5 索引的创建

#### 1. 单索引文件的建立

(1) 格式: INDEX ON <索引关键表达式> TO <索引文件名> [UNIQUE] FOR <条件> [ADDITIVE]

(2) 功能: 对当前表中满足条件的记录, 按<索引关键表达式>的值建立一个索引文件, 并打开此索引文件, 默认的文件扩展名为.idx。

(3) 参数描述。

<索引关键表达式>是用于指定记录重新排序的字段或表达式。<索引关键表达式>可以是字段名, 也可以是含有当前表中字段的合法表达式。表达式值的数据类型可以是字符型、数值型、日期型和逻辑型。若表达式中包含几种类型的字段名, 则常常需要使用类型转换函数将其转换为相同类型的数据。

[UNIQUE]。指定 UNIQUE 子句时, 若多条记录的<索引关键表达式>的值相同, 则只把第一次遇到的记录进行排序加入到索引文件中; 若省略该子句, 则把所有遇到的记录值都加入到索引文件中。

[ADDITIVE]。若省略 ADDITIVE 子句, 当为一个表建立新的索引文件时, 除结构复合索引文件外, 其他所有打开的索引文件都将被关闭; 若选择此项, 则已打开的索引文件仍然保持打开状态。

FOR <条件>。指定一个条件, 只显示和访问满足该条件表达式的记录, 索引文件只为那些满足条件表达式的记录创建索引关键字。

(4) 说明: 单索引文件总是按升序排列, 对于一个表文件, 允许建立多个索引文件。

#### 2. 复合索引文件的建立

复合索引文件是由索引标记组成的, 每个复合索引文件可包含多个索引标记, 每个索引标记都有标记名, 一个索引标记相当于一个单索引文件。

(1) 格式: INDEX ON <索引关键表达式> TAG <标记名> [OF <复合索引文件名>] [FOR <条件>] [ASCENDING | DESCENDING] [UNIQUE] [ADDITIVE]

(2) 功能: 建立和修改复合索引文件, 并打开此索引文件, 默认的文件扩展名为.cdx。

(3) 参数描述。

<索引关键表达式>、[FOR <条件>]、[ADDITIVE]与单索引文件的相同。

TAG <标记名> [OF <复合索引文件名>] 创建一个复合索引文件。当 TAG <标记名> 参数中不包含可选的[OF <复合索引文件名>]子句时, 便可以创建结构复合索引文件。如果 TAG <标记名> 参数后包含可选项[OF <复合索引文件名>]子句, 则可以创建非结构复合索引文件。

[ASCENDING|DESCENDING]中的 ASCENDING 指定复合索引文件为升序, 这是默认值, DESCENDING 指定复合索引文件为降序。

对于一个索引关键值, 只有第一个满足[UNIQUE]值的记录包含在.idx文件或.cdx 标志中。利用 UNIQUE 子句可以避免显示或访问记录的重重复值。

(4) 说明。

执行上述命令时, 系统先检查指定的复合索引文件是否存在, 若存在, 则在此文件

中增加一个索引标记；若不存在，则建立此索引文件。

标记名的命名规则与变量名的命名规则相同。

单索引文件只能按升序排列，而复合索引文件既可以按升序排列，也可以按降序排列，选择 ASCENDING 为升序排列，选择 DESCENDING 为降序排列。

表的显示和访问顺序只由一个索引文件（主控索引文件）和标志（主控标志）控制。有些命令使用主控索引文件和标志搜索记录，但是在修改表时，所有已打开的索引文件都将被更新。

从以上命令格式可以看出，使用命令可以建立普通索引、唯一索引和候选索引，但没有主索引。主索引需要在项目管理器中建立，我们将会在下一个任务中讲述。

### 3.1.6 索引的使用

建立一个索引后，如果不对它进行一系列的操作，它是不会对数据表起任何作用的，要让索引发挥作用，必须将其打开。

#### 1. 打开索引文件

结构复合索引文件在打开表时将被自动打开，但非结构的索引必须在使用之前打开索引文件。

格式：SET INDEX TO IndexFileList

其中 IndexFileList 是用逗号分开的文件列表，可以包含 .idx 索引和 .cdx 索引。

功能说明：虽然可以同时打开多个索引文件，但同一时间只能有一个单索引文件或复合索引文件中的一个索引标志作为主控索引，记录的操作和显示由主控索引来控制，如果<索引文件名表>中排在第一位的是单索引文件，则该单索引就是主控索引，如果排在第一位的是复合索引文件，则还需要用 ORDER 子句规定主控索引，如果不使用 ORDER 子句，记录仍将按记录号的顺序显示。

#### 2. 设置主控索引

如果在打开索引文件时未指定主控索引，打开索引文件后需要指定主控索引，或者希望改变主控索引，那么可使用下面的命令。

(1) 格式：SET ORDER TO [<数值表达式> <单索引文件名> [TAG] <索引标记> [OF <复合索引文件名>] [ASCENDING | DESCENDING]

(2) 功能：在打开的索引文件中指定主控索引。

(3) 参数描述。

<数值表达式>指定主控索引或标志的编号。数值表达式指定的是在 USE 或 SET INDEX 中列出的索引文件或标志的序号。

<单索引文件名>指定作为主控索引文件的单索引文件名。

[TAG] <索引标记> [OF <复合索引文件名>]指定复合索引文件中的一个索引标志为主控索引，标志名来自结构复合索引文件或任何打开的独立复合索引文件。如果在各个打开的独立复合索引文件中存在相同的标志名，应使用[OF <复合索引文件名>]来指定包含此标志的复合索引文件。

[ASCENDING | DESCENDING]参数的用法与前面的相同。



#### (4) 说明。

如果命令中不使用任何选项,只写 SET ORDER TO,那么系统不会指定主索引,记录仍将按照记录号的顺序显示,但并不关闭索引文件。

VFP 允许在 SET ORDER 命令中使用 ASCENDING 和 DESCENDING 暂时转换主控索引的顺序,但它不同于 SORT ON 命令中的 ASCENDING 和 DESCENDING。

当数值表达式的值为 0,或者省略所有可选项时,恢复表文件的自然顺序,但不关闭索引文件。

在打开的索引文件中,只有主控索引文件有效。

### 3. 索引文件的重建和更新

当表中的记录发生变化时,如果未打开索引文件,那么索引文件(结构复合索引文件除外)不会自动更新,要更新这些索引文件,可用下面的命令。

格式: REINDEX

功能: 重建当前打开的所有索引文件。

说明: 使用 REINDEX 命令之前,必须首先打开需要重建索引的全部索引文件,在重建索引时,使用包含 UNIQUE 关键字的 INDEX 命令或 SET UNIQUE ON 命令创建的索引文件,保持 UNIQUE 状态。

### 4. 快速查询

可对创建索引的表实现快速查找及定位。

#### (1) FIND 命令。

格式: FIND<字符型常量> <数值型常量>

功能: 在已经建立索引并且表文件和索引文件均已打开的表中,查找索引关键值与指定的字符串或数值型常量相匹配的第 1 条记录,如果找到了,则把记录指针指向该记录;如果没有找到,则将记录指针指向表的末尾。

#### (2) SEEK 命令。

格式: SEEK<表达式>

功能: 在打开的索引文件中快速查找与<表达式>相匹配的第 1 条记录。

参数描述: <表达式>指定 SEEK 搜索的关键字,<表达式>可以是空字符串。

说明: 当省略所有可选项时,SEEK 命令与 FIND 命令的功能基本相同,但 SEEK 命令的功能更强;SEEK 命令不仅可以查找字符串和常数,还可以查找字符型、数值型、日期型和逻辑型表达式的值;用 SEEK 命令查找字符串常量时,字符串常量必须放在定界符中。

SEEK 命令中的表达式必须和索引表达式的类型相同。

### 3.1.7 排序

排序是将关键字段值相同的记录按照顺序存放在一起,生成一个新的表文件。用于排序的字段称为关键字。

格式: SORT TO <表文件名> ON <字段名 1> [/A | /D] [/C] [,<字段名 2> [/A | /D] [/C] ...] [ASCENDING | DESCENDING] [<范围>] [FOR <逻辑表达式>] [WHILE <逻辑表达式>] [FIELDS <字段名列表>]

功能: 对当前选定的表排序,并将排序后的记录输出到新表中。

参数描述如下。

(1) <表文件名>指定经过排序后生成的新表的表文件名。

(2) ON <字段名 1>在当前选定的、要排序的表中指定关键字段，字段的内容和数据类型决定了记录在新表中的顺序。

(3) [/A|/D] [/C]指定排序顺序（升序或降序）。"/A"指定为按升序排序，"/D"指定按降序排序。如果字符型字段名后面包含"/C"，则忽略大小写。可以把"/C"选项与"/A"或"/D"选项组合起来。

### 3.1.8 排序与索引的区别

从上面的讲述我们可以看出，排序与索引之间存在着一定的联系，但是，我们还要看到它们之间的区别，区别主要有以下几个方面。

(1) 排序要生成一个新的表文件，记录的物理顺序发生了改变。

(2) 排序生成的表可以单独使用。

(3) 索引并不生成新的表文件，仅仅是表中记录的逻辑顺序发生了变化，但索引会生成一个新的文件，即索引文件。

(4) 索引文件不能单独使用，它必须同表一起配合使用。

### 3.1.9 数据完整性

在数据库中数据完整性是指保证数据正确的特性。数据完整性一般包括实体完整性、域完整性和参照完整性等。Visual FoxPro 提供了实现这些完整性的方法和手段。

#### 1. 实体完整性

实体完整性是保证表中记录唯一的特性，即在一个表中不允许有重复的记录。在 Visual FoxPro 中利用主关键字或候选关键字来保证表中的记录唯一，即保证实体唯一性。

如果一个字段的值或几个字段的值能够唯一标志表中的一条记录，则将这样的字段称为候选关键字。一个表中可能会有几个具有这种特性的字段或字段的组合，这时从中选择一个作为主关键字。在 Visual FoxPro 中将主关键字称为主索引，将候选关键字称为候选索引。由上所述，在 Visual FoxPro 中主索引和候选索引具有相同的作用。

#### 2. 域完整性

我们以前所熟知的数据类型的定义属于域完整性的范畴。例如，对于数据值字段，通过指定不同的宽度和小数位数说明不同范围的数值数据类型，从而可以限定字段的取值类型和范围。但这些对域完整性远远不够，还可以用一些域约束规则（也称为字段有效性规则）来进一步保证域完整性，在插入或修改字段值时激活，主要用于数据输入正确性的检验。

域约束规则需要在表设计器中建立，我们将在下一个任务中讲述设置域完整性约束的操作步骤。

#### 3. 参照完整性

参照完整性与表之间的联系有关，其含义是，当插入、删除或更新一个表中的数据时，通过参照引用相互关联的另一个表中的数据，来检查对表的数据的操作是否正确。假如一



个学生成绩表中有“学号”、“课程编号”、“成绩”字段，当插入一条记录，例如，学号：0104111；课程编号：yy0005；成绩：90。如果没有参照完整性检查，则可能会插入一个“学号”或“课程编号”并不存在的记录，这时插入的记录肯定是错误的。如果在插入学号、课程编号记录之前，能够进行参照完整性检查，检查指定记录在表中是否存在，则可以保证插入记录的合法性。

参照完整性是关系数据库管理系统的一个很重要的功能。在 Visual FoxPro 中，为了建立参照完整性，必须首先建立表之间的联系。

最常见的联系类型是一对多的联系，在关系数据库中通过连接字段来体现和表示联系。连接字段在父表中一般是主关键字，在子表中是外部关键字。如果一个字段或字段的组合不是本表的关键字，而是另外一个表的关键字，则这样的字段称为外部关键字。

## 任务 3.2 实例训练

### 3.2.1 利用表设计器创建“课程表”

创建一个新表时，关键的一步是要确定表中的字段，即定义表的结构。例如，我们要建立学生学籍管理系统中的“课程表”，假设表文件名为 objects.dbf，对“课程表”的各字段的描述如表 3-1 所示。

表 3-1 课程表

| 课程编号<br>(C6) | 课程名称<br>(C20) | 开设学期<br>(C10) | 任课教师<br>(C8) | 课时<br>(N2) | 学分<br>(N2) | 选修或必修<br>(C4) |
|--------------|---------------|---------------|--------------|------------|------------|---------------|
| jsj001       | 操作系统          | 第一学期          | 赵玉           | 50         | 5          | 必修            |
| jsj002       | VB 语言         | 第一学期          | 钱红           | 45         | 4          | 必修            |
| jsj003       | 计算机基础         | 第一学期          | 孙南           | 45         | 2          | 必修            |
| jsj004       | 计算机网络         | 第二学期          | 郑东           | 40         | 3          | 选修            |
| jsj005       | 面向对象技术        | 第三学期          | 李兰           | 45         | 4          | 必修            |
| jsj006       | Delphi 应用     | 第三学期          | 周五           | 50         | 3          | 必修            |

下面我们利用表设计器创建表的结构，操作步骤如下。

(1) 在菜单栏中选择“文件”|“新建”命令（或在“常用”工具栏上单击“新建”按钮），弹出“新建”对话框，如图 3-1 所示。

(2) 在“新建”对话框中选中“表”单选按钮选项，单击“新建”按钮，弹出“创建”对话框。

(3) 在“创建”对话框中选择路径，输入表文件名 objects，单击“保存”按钮，弹出“表设计器”对话框，如图 3-2 所示，在表设计器的“字段”选项卡中输入各字段信息。

(4) 表设计器包括 3 个选项卡：“字段”、“索引”和“表”。其中，“字段”选项卡显示各个字段的名称、类型和长度；“索引”选项卡可创建索引；“表”选项卡可显示表的状态。

在“字段名”列中输入字段名，如课程编号、课程名称、开设学期、任课教师等。

在“类型”列中，选择下拉列表中的某一字段类型，分别有：字符型、货币型、数值型、浮点型、日期型、日期时间型、双精度型、整型、逻辑型、备注型、通用型、字符型（二进制）、备注（二进制）等。需要注意的是，字段的数据类型应与字段中将要存储信息的数据类型相匹配，如果“类型”是“数据值”或“浮点型”，要设置“小数位数”数值框中的小数点位数。



图 3-1 “新建”对话框

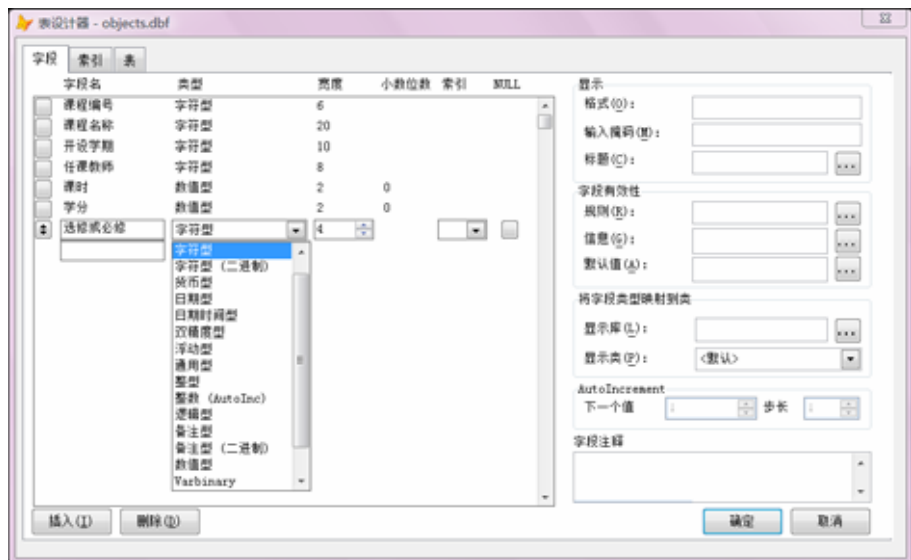


图 3-2 “表设计器”对话框

在“宽度”列中，设置以字符（字节）为单位的列宽，使字段的宽度足够容纳将要显示的信息内容。需要注意的是，一个汉字占 2 字节。

如果希望为字段添加索引，就在“索引”列中选择一种排序方式。

如果想让字段接受 NULL 值，则选中 NULL 选项，NULL 就是默认值或无确定的值，它不等于零或空格，不能把它理解为任何意义的数据。一个 NULL 值不能与某个值（包括另一个 NULL 值）比大小。字段名前的双向箭头表明是当前行，同一行的各项目之间用 Tab 键移动。

在右边的“显示”栏中，指定该字段的输入和显示格式属性。

- 格式：为在浏览窗口、表单或报表中该字段的大小写、尺寸和显示样式指定表达式，表单和报表中的属性设置可以覆盖该表达式。



- 输入掩码 :为插入到该字段的值指定格式 ,如电话号码可能拥有( 999 )999-9999 的格式。
- 标题 :指定在浏览窗口、表单或报表中该字段的出现标签。表单和报表中的属性设置可以覆盖该设置。要为字段标题指定一个表达式 ,可在标题串前面包含一个等号 (=)。表达式不能超过 254 个字符 ,如果超过了 254 个字符 ,则标题被默认为字段名。

在“字段有效性”栏中,规则指定强制数据有效性的字段级规则。信息指定一个输入违反了字段有效性规则时要显示错误的信息。“默认值”指定该字段的默认值。

在“将字段类型映射到类”栏中,要指定默认控件类,设置将字段拖放到表单时要创建的库和类。“显示库”指定类库的路径和文件名。要浏览文件,可单击其后面的按钮。“显示类”指定该字段的默认控件类。

在“AutoIncrement (自动增量)”栏中,设置字段启用自动增量之后的起始和增量值,在“下一值”和“增量”数值框中选择值。

在“字段注释”栏中输入数据库表中的字段的描述,易于了解和更新表中所有字段的属性定义完成后,单击“确定”按钮,则结束表结构的定义。

定义新表结构后,新建立的表还只是一个空表,表中没有保存任何记录数据。为此,还需要将每门课程的信息输入到表中。表中记录的添加、浏览等内容我们将在后面讲到。

### 3.2.2 在“学籍管理数据库”下创建“学生信息表”

创建数据库表有以下 3 种方法。

#### 1. 利用新建表创建新表

操作步骤如下。

在学生学籍管理系统项目“学生学籍管理系统.pjx”管理器中的“学籍管理数据库”下创建学生信息表 students.dbf 的表结构。

- (1) 打开项目管理器“学生学籍管理系统.pjx”。
- (2) 选择“数据”选项卡,展开“数据库”选项。
- (3) 展开“学籍管理数据库”,并选择“表”选项。

以上三步操作如图 3-3 所示。



图 3-3 操作示意图

(4) 单击“新建”按钮,弹出“新建表”对话框,如图 3-4 所示,单击“新建表”按钮。

(5) 弹出如图 3-5 所示的“创建”对话框,输入新表名称:students.dbf。



图 3-4 “新建表”对话框



图 3-5 “创建”对话框

(6) 单击“保存”按钮后,即可打开如图 3-6 所示的“表设计器”对话框。

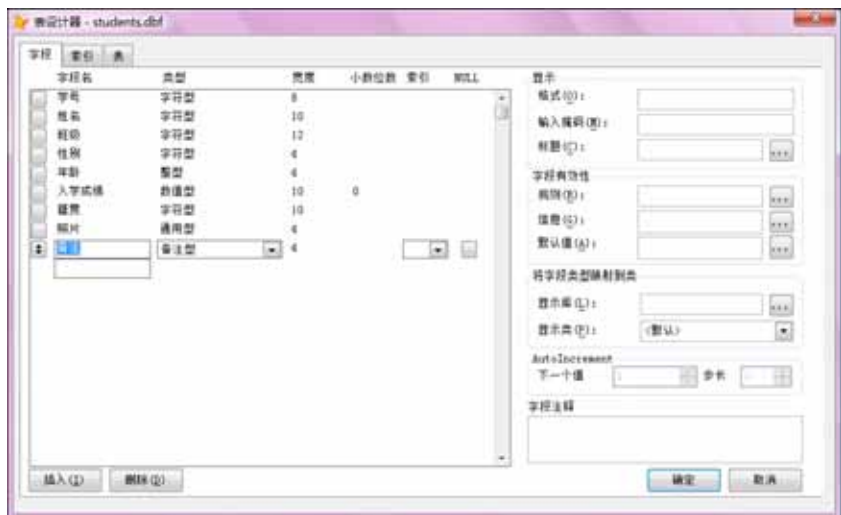


图 3-6 “表设计器”对话框

(7) 在表设计器的“字段”选项卡中输入各字段信息。

## 2. 利用表向导创建新表

利用表向导创建新表,就是把已有的表作为样本,在表向导的引导下,逐步完成创建新表的过程。

例如,在学生学籍管理系统中,在学生信息表 students.dbf 的基础上,利用表向导创建一个名为 stud 的表,要求表中包含“学号”、“姓名”、“班别”字段。

操作步骤如下。

(1) 在项目管埋器的“数据”选项卡中选择“自由表”选项,单击“新建”按钮,弹出“新建表”对话框,单击“表向导”按钮,弹出“表向导”的第一步对话框。也可在菜





单栏上选择“文件”|“新建”命令,弹出“新建”对话框,选中“表”单选按钮,单击“向导”按钮,弹出“Table Wizard (表向导)”的第一步对话框,如图 3-7 所示。

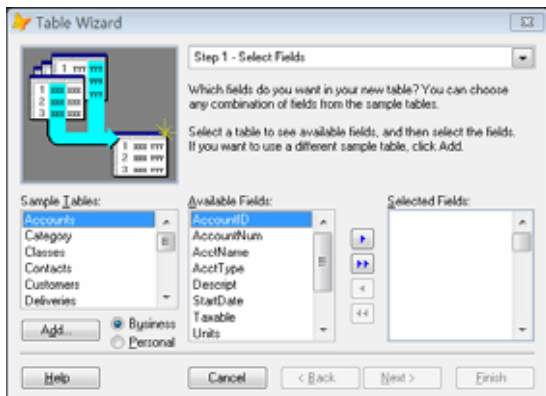


图 3-7 “Table Wizard (表向导)” 第一步对话框

(2) 在“Sample Table (样本表)”列表框中选择可作为样表的表。如果没有可作为样表的表,可单击“Add(添加)”按钮,弹出“打开”对话框,输入可作为样表的表 students.dbf,单击“Add”按钮,将选中的表添加到“Sample Table (样本表)”列表框中。(注意:一般来说,在我们第一次打开“表向导”对话框时,会发现在“Sample Table”中有一个 students 表,但是这个表并不是我们所创建的 students 表,这个表的字段和我们所创建的表的字段不同,因此我们仍然需要添加自己创建的 students 表作为样本表。)

这时,“Available Fields (可用字段)”列表框中将显示表中的全部字段,用户可从中选择所需要的字段。

选择字段时,可以用“Available Fields (可用字段)”列表框右边的 4 个按钮进行操作。这 4 个按钮的功能如下:

- ▶ 选取向导表中某个字段;
- ▶▶ 选取向导表中所有字段;
- ◀ 删除向导表中某个字段;
- ◀◀ 删除向导表中所有字段。

根据题目的要求,逐一选择新表中的字段。当所需要的字段全部被选定后,“Selected Fields (选定字段)”列表框中将显示已选定的字段名,如图 3-8 所示。

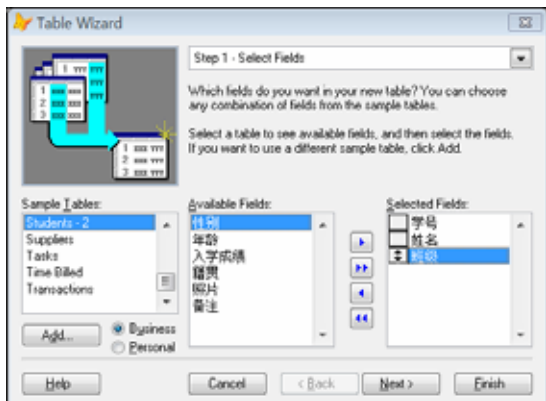


图 3-8 “选定字段”列表框中显示已选定的字段名

(3) 单击“Next (下一步)”按钮,在“Table Wizard (表向导)”第一步(a)对话框中,确定新表为自由表,如图 3-9 所示。

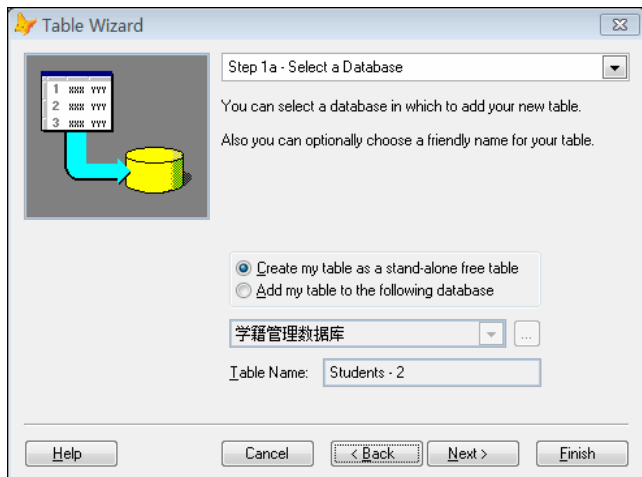


图 3-9 “Table Wizard (表向导)” 第一步(a)对话框

(4) 单击“Next (下一步)”按钮,在“Table Wizard (表向导)”第二步对话框中,可对已选定的字段进行 Field Name (字段名)、Type (字段类型)、Width (字段长度)的修改和确认,如图 3-10 所示。

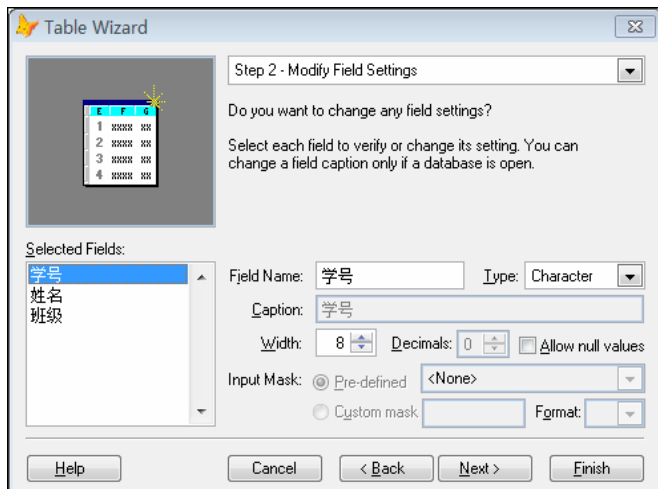


图 3-10 “Table Wizard (表向导)” 第二步对话框

(5) 单击“Next (下一步)”按钮,在“Table Wizard (表向导)”第三步对话框中,可为表建立索引,如图 3-11 所示。

(6) 单击“Next (下一步)”按钮,在“Table Wizard (表向导)”第四步对话框中,有 3 个供用户选择的单选按钮,如图 3-12 所示,其功能如下。

Save table for later use (保存表为以后使用): 只保存表。

Save table and browse it (保存表并浏览它): 保存表,并进入表浏览窗口。

Save table and modify it in the Table Designer (保存表并在表设计器中修改它): 保存表,并进入表设计器窗口。

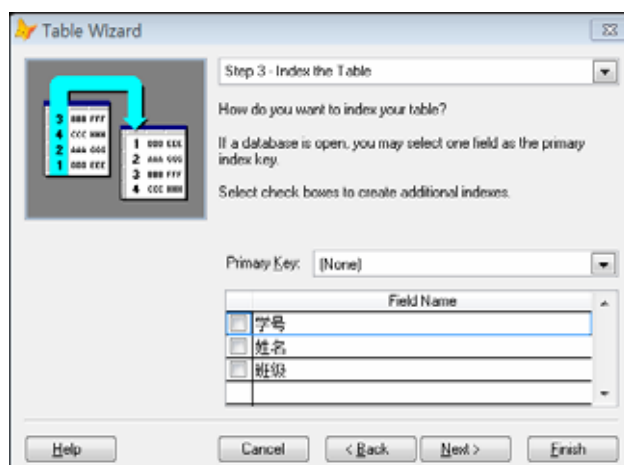


图 3-11 “Table Wizard (表向导)” 第三步对话框

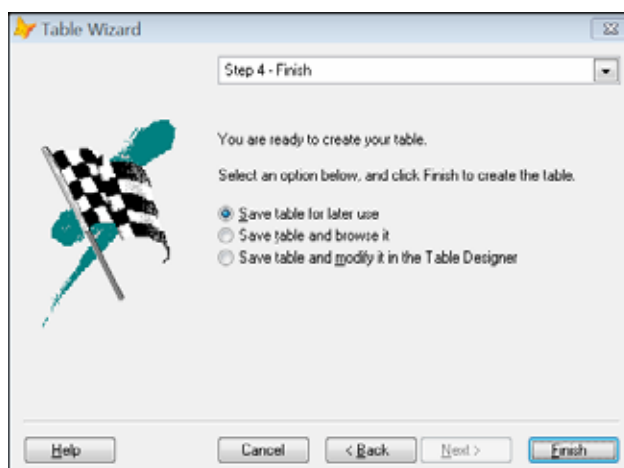


图 3-12 “Table Wizard (表向导)” 第四步对话框

(7) 单击“Finish (完成)”按钮，弹出“另存为”对话框，如图 3-13 所示。输入新建的表名 stud，然后单击“保存”按钮，完成该新表文件的创建。



图 3-13 “另存为”对话框

### 3. 使用命令建立表结构

命令格式：CREATE<表文件名>

功能：创建一个以<表文件名>为名的新表。

例如，以下命令可以打开表设计器，创建表文件——成绩表 scores。

CREATE SCORES

#### 3.2.3 修改“学生信息表”的结构

##### 1. 修改表的结构

(1) 可以用以下 3 种方法打开表设计器，对表结构进行修改。

用项目管理器打开表设计器。

在项目管理器中选定要修改的数据表，单击“修改”按钮，选定的表结构将显示在表设计器中。

用菜单方式打开表设计器。

先打开选中的表文件，选择“显示”“表设计器”命令，即可弹出“表设计器”对话框。

用命令方式打开表设计器。

命令格式：MODIFY STRUCTURE

命令功能：修改当前表的结构。

说明：如果没有确定当前表，可使用 USE 命令打开一个表；该命令的功能为打开以<表文件名>为名的表，格式为 USE [<表文件名>]

(2) 案例说明。

例如，修改“学籍管理数据库”中表 students.dbf 的“性别”字段的宽度，由 4 改为 2。操作步骤如下。

选择“文件”“打开”命令，弹出“打开”对话框，如图 3-14 所示，选择项目“学生学籍管理系统.pjx”。



图 3-14 “打开”对话框



单击“确定”按钮打开项目管理器，选择数据表 students，单击“修改”按钮，打开表设计器，如图 3-15 所示。



图 3-15 表设计器

在表设计器中，修改“性别”字段的宽度，如图 3-16 所示。



图 3-16 修改“性别”字段的宽度

表结构修改好以后，单击“确定”按钮，弹出系统提示对话框，如图 3-17 所示。

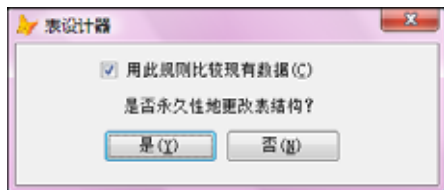


图 3-17 系统提示对话框

在系统提示对话框中，单击“是”按钮，确认修改后的表结构。

## 2. 在表中增加字段

在表设计器中单击“插入”按钮，在“字段名”列中输入字段名，在“类型”列中选择字段的数据类型，在“宽度”列中设置或输入字段宽度。修改完成后，单击“确定”按钮，弹出系统提示对话框，单击“是”按钮确认修改后的表结构。

例如，在学生信息表 students.dbf 中增加“入学时间”字段（日期时间型），如图 3-18 所示。



图 3-18 在表设计器中增加“入学时间”字段

## 3. 删除表中的字段

在表设计器中选定要删除的字段，单击“删除”按钮即可。

## 4. 改变字段顺序

如果需要改变某字段的顺序，只要选中该字段最左边的空格，拖动鼠标上、下移动，直到合适位置释放鼠标即可。

### 3.2.4 显示“学生信息表”记录

在使用表的过程中，经常需要对表中的记录进行编辑，本节我们将介绍表中记录的添加与显示。

#### 1. 打开数据库表

(1) 从项目管理器中打开数据库表。操作步骤如下。

打开项目管理器，双击“数据库”选项。

选择“学籍管理数据库”，双击“表”选项。

选择学生信息表 students，如图 3-19 所示。

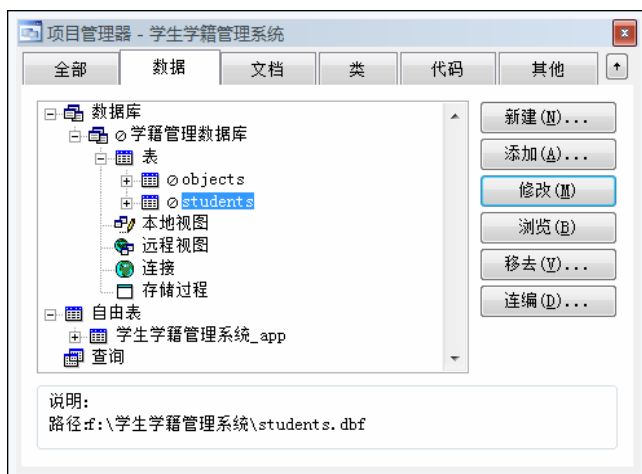


图 3-19 打开学生信息表 students

单击“浏览”按钮，就可显示当前表记录的内容了。

(2) 从菜单栏打开数据库表。操作步骤如下。

单击“文件” “打开”命令。

弹出“打开”对话框，选择所需要打开表文件的位置，即选择盘符、路径及表名。需要注意的是，打开某个表后屏幕上并没有显示。

选择“显示” “浏览”命令，即可将当前工作区中已打开的表文件内容显示在浏览窗口中。

## 2. 往表中添加记录

例如，往学生信息表 students 表中添加记录。

操作步骤如下。

(1) 在项目管理器中点击 students 表，单击“浏览”按钮，然后选择“显示” “编辑”命令，即打开编辑模式。

(2) 在命令栏中输入 insert 命令，然后按键盘上的 Enter 键，即出现如图 3-20 所示的添加窗口。

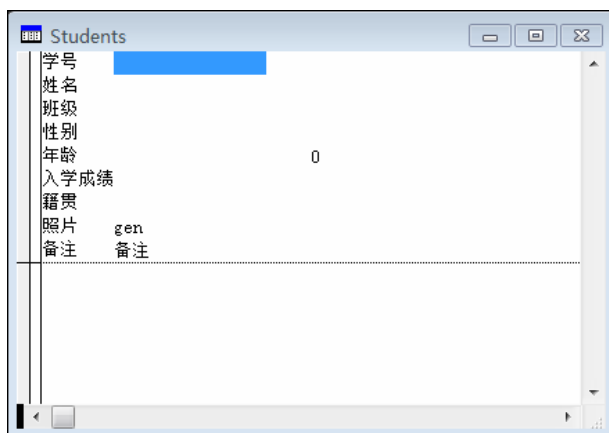


图 3-20 为 students 表添加记录

- (3) 添加每个学生的各项信息。
- (4) 记录添加完后，单击右上角的“关闭”按钮即自动保存。

### 3. 显示表记录

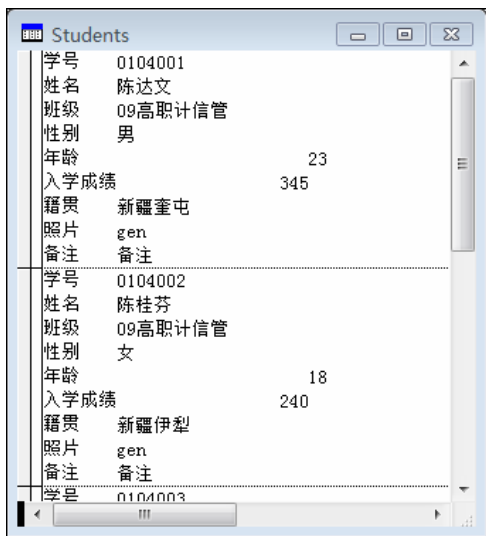
- (1) 浏览、编辑模式。

显示的表记录有两种模式：一种是浏览模式，如图 3-21 所示；另一种是编辑模式，如图 3-22 所示。这两种模式可以通过“显示”菜单下的“浏览”、“编辑”命令进行切换。



| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  | 备注 |
|---------|-----|---------|----|----|------|------|-----|----|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | 345  | 新疆奎屯 | gen | 备注 |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240  | 新疆伊犁 | gen | 备注 |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420  | 广东高州 | gen | 备注 |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380  | 广东湛江 | gen | 备注 |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290  | 广东徐闻 | gen | 备注 |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315  | 广东湛江 | gen | 备注 |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300  | 广东廉江 | gen | 备注 |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 | gen | 备注 |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310  | 广东揭阳 | gen | 备注 |

图 3-21 浏览模式下的 students 表



|      |         |
|------|---------|
| 学号   | 0104001 |
| 姓名   | 陈达文     |
| 班级   | 09高职计信管 |
| 性别   | 男       |
| 年龄   | 23      |
| 入学成绩 | 345     |
| 籍贯   | 新疆奎屯    |
| 照片   | gen     |
| 备注   | 备注      |
| 学号   | 0104002 |
| 姓名   | 陈桂芬     |
| 班级   | 09高职计信管 |
| 性别   | 女       |
| 年龄   | 18      |
| 入学成绩 | 240     |
| 籍贯   | 新疆伊犁    |
| 照片   | gen     |
| 备注   | 备注      |
| 学号   | 0104003 |

图 3-22 编辑模式下的 students 表

注意：浏览和编辑这两种模式也可以同时显示。

操作步骤如下。

- 将鼠标移动到编辑模式或浏览模式下窗口左下角的黑色小长方形处。
- 按住鼠标并向右拖动，则该窗口被分成两栏。





这两栏可以同时显示编辑模式和浏览模式，也可以一边显示编辑模式，一边显示浏览模式，如图 3-23 所示。

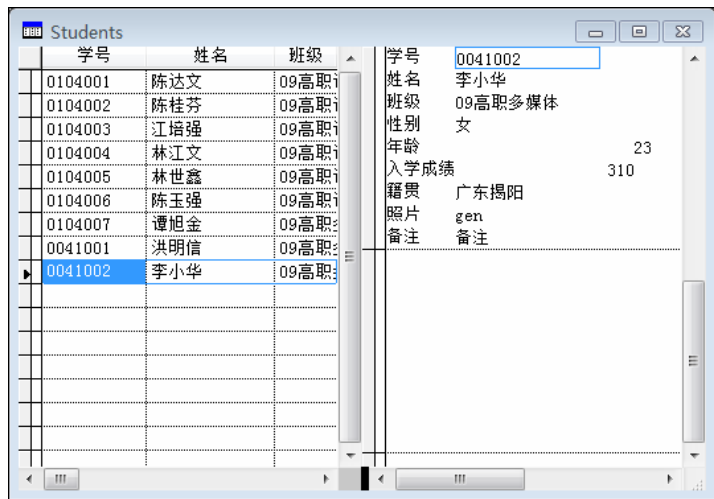


图 3-23 浏览、编辑模式下的 students 表

## (2) 命令方式。

除了上面两种方式外，还可以使用 LIST 或 DISPLAY 命令显示记录。

命令格式：

LIST|DISPLAY [ FIELDS <字段名列表> ] [ <范围> ] [ FOR<条件>/WHILE<条件> ] [ OFF ]

功能：显示当前表中的全部或部分记录和数据。

参数描述如下。

[OFF]：使用 OFF 时，不显示记录号。

[<范围>]：为可选项，选择时为 ALL、RECORD n、NEXT n、REST 中的一个参数，表示记录显示的范围。其中 ALL 为所有记录，RECORD n 为第 n 号记录，NEXT n 是从当前记录开始的 n 条记录，REST 为从当前记录开始的所有剩余记录。

FIELDS <字段名列表>：若省略<字段名列表>，则显示当前表中的所有字段，否则显示指定的字段。如果备注字段名出现在<字段名列表>中，则它的内容按 50 个字符列宽显示。

[FOR<条件>/WHILE<条件>]：该子句用于有选择地显示某些记录，省略时则显示<范围>限定的全部记录。

LIST 和 DISPLAY 的区别。DISPLAY 每显示一屏记录就暂停一次，按任意键后继续显示剩余的记录，而 LIST 没有周期性暂停，连续向下显示，直到记录显示完毕为止；若省略所有可选项，则 DISPLAY 命令显示当前记录，即范围为 NEXT 1，而 LIST 命令显示全部记录，即范围为 ALL。

例如，显示课程表 objects 中第一学期开设的课程名称、开设学期、任课教师和课时字段的内容。

在命令栏中输入命令：

```
USE OBJECTS &&打开表 objects
LIST FIELDS 课程名称,开设学期,任课教师,课时 FOR 开设学期="第一学期"
```

LIST 屏幕显示结果如图 3-24 所示。

| 记录号 | 课程名称  | 开设学期 | 任课教师 | 课时 |
|-----|-------|------|------|----|
| 1   | 操作系统  | 第一学期 | 赵玉   | 50 |
| 2   | VB语言  | 第一学期 | 钱红   | 45 |
| 3   | 计算机基础 | 第一学期 | 孙南   | 45 |

图 3-24 LIST 屏幕显示结果

再如，显示学生信息表 students 中第 2 条记录的所有字段的内容。  
在命令栏中输入命令：

|              |                |
|--------------|----------------|
| USE STUDENTS | &&打开表 students |
| GO 2         | &&第 2 条记录为当前记录 |
| DISPLAY      | &&显示当前记录       |

DISPLAY 屏幕显示结果如图 3-25 所示。

| 记录号 | 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  | 备注 |
|-----|---------|-----|---------|----|----|------|------|-----|----|
| 2   | 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240  | 新疆伊犁 | gen | 备注 |

图 3-25 DISPLAY 屏幕显示结果

#### 4. 调整浏览窗口的行和列

浏览窗口可根据用户的需要，进行多种界面的调整变化，变得更加符合用户的习惯。

(1) 改变数据表的大小。通过鼠标的拖动操作，可以随心所欲地调整浏览窗口。

(2) 调整表行高。在 Visual FoxPro 中，一旦改变了第一行的高度后，整个表行也随之改变。显然，只要改变第一行的高度即可。

操作步骤如下。

将鼠标指针移到行标头上的第一条记录与第二条记录之间的分隔线上，当鼠标指针变为如图 3-26 所示的“十”字形时，上、下拖动鼠标改变整个浏览窗口中记录的行高，直到合适位置为止。



图 3-26 改变行高



(3) 调整列宽。方法与改变行高相同，只是将鼠标指针移到某一列的列标头上的合适位置处，当鼠标指针变为如图 3-27 所示的“十”字形时，左、右拖动鼠标改变当前列宽，直到合适位置为止。

| 学号      | 姓名  | 班级      | 性别 | 年龄 |
|---------|-----|---------|----|----|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 |

图 3-27 改变列宽

(4) 移动字段显示位置。在浏览窗口中，字段的相对位置是根据用户建立表结构时的顺序排列的，但用户可根据需要来改变这种排列方式。

操作步骤如下。

将鼠标指针移到需要移动的字段的列标头处，此时，鼠标指针变为向下箭头形式，且该字段名变为黑色，拖动鼠标到列标头的新位置后，该字段位置将随之改变。

如图 3-28 所示是改变字段当前显示位置，将原来学生信息表 students 中的“班级”字段移动到了第二列，“学号”字段移动到了第三列。要注意的是，这种方式只能改变其当前显示位置，表结构中的字段位置仍然没有改变。

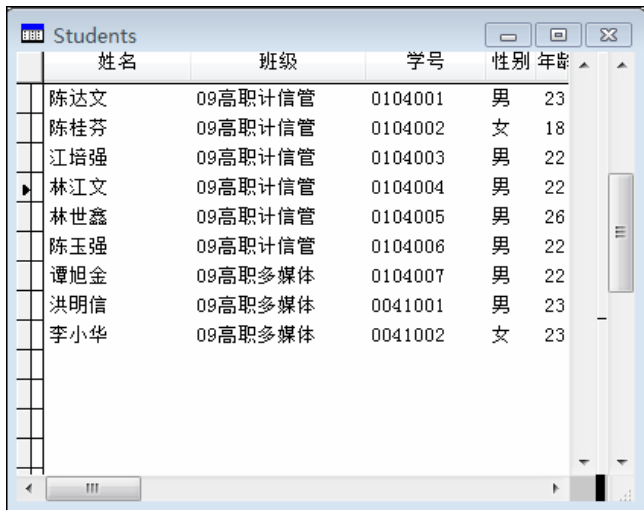
| 姓名  | 班级      | 学号      | 性别 | 年龄 |
|-----|---------|---------|----|----|
| 陈达文 | 09高职计信管 | 0104001 | 男  | 23 |
| 陈桂芬 | 09高职计信管 | 0104002 | 女  | 18 |
| 江培强 | 09高职计信管 | 0104003 | 男  | 22 |
| 林江文 | 09高职计信管 | 0104004 | 男  | 22 |
| 林世鑫 | 09高职计信管 | 0104005 | 男  | 26 |
| 陈玉强 | 09高职计信管 | 0104006 | 男  | 22 |
| 谭旭金 | 09高职多媒体 | 0104007 | 男  | 22 |
| 洪明信 | 09高职多媒体 | 0041001 | 男  | 23 |
| 李小华 | 09高职多媒体 | 0041002 | 女  | 23 |

图 3-28 改变字段当前显示位置

(5) 去除网格线。从图 3-28 中可以看出,在浏览窗口中,每个记录,每个字段之间都有虚线,这称为网格线。这些网格线是可以设置的,用户可以根据需要确定是否显示。

操作步骤如下。

选择“显示”“网格线”命令,使该命令前面的“ ”号消失。如图 3-29 所示的浏览窗口是去除网格线后的格式,若需要恢复,再次单击该命令即可。



| 姓名  | 班级      | 学号      | 性别 | 年龄 |
|-----|---------|---------|----|----|
| 陈达文 | 09高职计信管 | 0104001 | 男  | 23 |
| 陈桂芬 | 09高职计信管 | 0104002 | 女  | 18 |
| 江培强 | 09高职计信管 | 0104003 | 男  | 22 |
| 林江文 | 09高职计信管 | 0104004 | 男  | 22 |
| 林世鑫 | 09高职计信管 | 0104005 | 男  | 26 |
| 陈玉强 | 09高职计信管 | 0104006 | 男  | 22 |
| 谭旭金 | 09高职多媒体 | 0104007 | 男  | 22 |
| 洪明信 | 09高职多媒体 | 0041001 | 男  | 23 |
| 李小华 | 09高职多媒体 | 0041002 | 女  | 23 |

图 3-29 去除网格线后的浏览窗口

### 3.2.5 浏览和修改“学生信息表”中的记录

在 Visual FoxPro 系统中,可采用“浏览”和“编辑”两种操作方法对表中的数据进行修改。

#### 1. 菜单方式

(1) 在浏览窗口中修改数据。

(2) 在编辑窗口中修改数据。

其操作与显示表记录中的浏览、编辑模式相同。

#### 2. 命令方式

(1) 编辑命令。

命令格式: EDIT [FIELDS<字段名表>][FOR<条件>]

CHANGE [FIELDS<字段名表>][FOR<条件>]

功能: 打开编辑窗口。

说明: 按给定条件打开表的编辑窗口。

(2) 浏览命令。

命令格式: BROWSE [FIELDS <字段名表>] [FOR <逻辑型表达式>]

功能: 在屏幕上打开一个浏览窗口,在窗口中显示表的记录。



参数描述。

[FIELDS <字段名表>]：指定在浏览窗口中显示的表的字段。

[FOR <逻辑型表达式>]：指定在浏览窗口中显示的记录所要求满足的条件。

说明：BROWSE 命令可以带有很多任选项，命令格式中只介绍了 BROWSE 命令的最基本的任选项。

(3) 记录的修改。

打开数据表后可用 REPLACE 命令成批修改字段值。REPLACE 是一条非全屏幕编辑命令。

命令格式：REPLACE <字段名 1> WITH <表达式 1> [ADDITIVE] [, <字段名 2>, <表达式 2> [ADDITIVE]]...[<范围>][FOR<条件>][WHILE<条件>]

功能：不进入全屏幕编辑方式，根据命令中指定的条件和范围，用表达式的值去更新指定字段的内容。

参数描述。

<字段名 1>：指定要替换值的字段。

WITH <表达式 1>：指定用来进行替换的表达式或值。

<范围>：指定进行替换时的记录范围。

FOR <条件>：指定要替换字段值的记录应满足的条件。

WHILE <条件>：当记录的条件不满足时结束替换，用于按条件中的字段建立了索引的表。

说明：WITH 后面的表达式的类型必须与 WITH 前面的字段类型一致。

例如，在学生信息表 students 中，每学年后年龄增长一岁。

命令如下：

```
USE STUDENTS
LIST
REPLACE ALL 年龄 WITH 年龄+1
LIST
```

### 3.2.6 向“学生信息表”中追加记录

在建立表结构时，如果没有用上述方法向表中添加数据，也可以用追加记录的方式向表中输入数据。

假设已打开学生信息表文件 students，在数据表中追加新记录的操作方法有以下 4 种。

#### 1. 用项目管理器打开浏览窗口追加记录数据

操作步骤如下。

(1) 在项目管理器中选择表名，单击“浏览”按钮，打开编辑窗口或浏览窗口。

(2) 在菜单栏中选择“显示”|“追加模式”命令，在最后一个空记录中输入新的数据，这种方法可以一次追加多条记录；也可以选择“表”|“追加新记录”命令，不过这种方法一次只能在表文件尾部追加一条新记录。编辑窗口如图 3-30 所示，浏览窗口如图 3-31 所示。



图 3-30 编辑窗口



图 3-31 浏览窗口

## 2. 用菜单方式打开编辑窗口或浏览窗口追加数据

操作步骤如下。

- (1) 在系统菜单栏上选择“文件”|“打开”命令，弹出“打开”对话框。
- (2) 在“文件类型”下拉列表框中选择“表(.dbf)”类型，再在文件列表框中选择所要打开的文件（如学生信息表 students），单击“确定”按钮。
- (3) 在系统菜单栏上选择“显示”|“编辑”命令，弹出编辑窗口，或者在系统菜单栏上选择“显示”|“浏览”命令，弹出浏览窗口。
- (4) 在系统菜单栏上选择“显示”|“追加模式”命令，可在最后一个空记录中输入新的数据。

## 3. 用命令方式追加记录数据

格式：APPEND [BLANK]

功能：在表的末尾添加一个或多个新记录。

参数描述：[BLANK]，在当前表的末尾添加一条空记录。

说明如下。

- (1) 可以在只有表结构而没有记录的空表中添加记录，也可以在已经录入数据的表中的尾部追加记录。
- (2) 可以向当前表的尾部追加一条或多条新记录。省略可选项，则在当前表的尾部追加任意新记录，追加记录的方法及操作与用 CREATE 命令建立表时输入数据的方法相同。
- (3) 当发出 APPEND 或 APPEND BLANK 命令，并且选定工作区中没有打开的表时，将显示一个“打开”对话框，可以在对话框中选择一个要添加记录的表。APPEND 命令打开一个编辑窗口，可以在其中输入一个或多个新记录。增加新记录后，VFP 将自动修改打开的所有索引。

例如，向课程表 objects 中追加若干条记录。

输入命令（见图 3-32）：

```
USE OBJECTS SHARED
APPEND
```



弹出如图 3-33 所示的追加记录窗口，可以追加若干条记录。

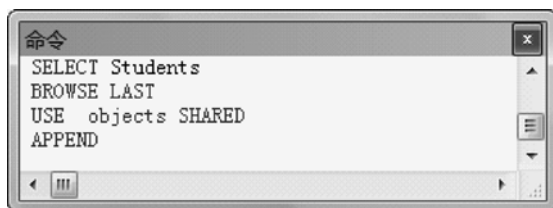


图 3-32 输入命令

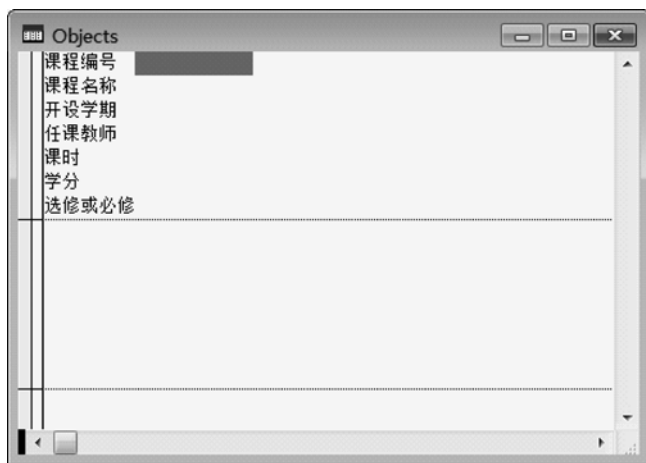


图 3-33 追加记录窗口

#### 4. 从另一个表文件成批追加记录

##### (1) 菜单方式。

操作步骤如下。

打开表的浏览窗口或编辑窗口后，在菜单栏上选择“表”|“追加记录”命令，弹出“追加来源”对话框，如图 3-34 所示。在该对话框中指出追加数据的来源（文件名、文件类型及该文件所在的文件夹等），并将其追加到当前打开的表文件中。若单击“选项”按钮，则弹出“追加来源选项”对话框，如图 3-35 所示，可以进一步设置数据源中可以被追加到当前表文件中的记录的条件，以及选择被追加的字段。



图 3-34 “追加来源”对话框

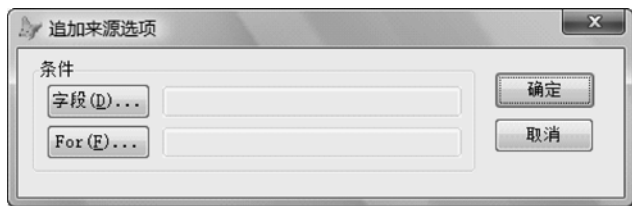


图 3-35 “追加来源选项”对话框

#### (2) 命令方式。

命令格式：APPEND FROM <表文件名> [FOR<条件表达式>] [FIELDS<字段名表>]

功能：从指定表文件中将满足条件的记录追加到当前表文件的末尾。

参数描述：表文件名是含有要追加数据的表文件名；可选项 FIELDS<字段名表>指明要追加的字段，字段名用逗号分开。

说明：若源文件中的字段比当前表文件中的字段长，则截取超出的部分，数值型字段则输入字符“\*”。

### 3.2.7 往“学生信息表”中插入记录

追加方式是在数据表文件的末尾增加若干条记录，而插入记录可以在任何位置增加新记录。

#### (1) 命令格式：INSERT[BLANK] [BEFORE]

#### (2) 命令功能：在当前记录之前或之后插入一条或多条新记录。

#### (3) 参数描述。

选择[BEFORE]子句，新记录插在当前记录之前，当前记录和其后的记录顺序向后移动；否则插在当前记录之后，当前记录之后的记录顺序向后移动。

选择[BLANK]子句，则插入一条空记录。

#### (4) 说明。

省略所有可选项，则在当前记录之后插入新记录。

在建立表结构之后，可以用 INSERT 命令向表中输入数据。

在 VFP 中，如果数据库具有表缓冲或行缓冲功能，则 INSERT 命令不能用于该数据库的表。对于具有参照完整性规则的表也不能使用 INSERT 命令。

### 3.2.8 删除“学生信息表”中的一些记录

在表的使用过程中，如果不需要表中的一些数据，可以将其暂时或永久地清除掉。

#### 1. 逻辑删除数据表中的记录

逻辑删除数据表中的记录就是给暂时不使用的记录添加一个删除标志。此外，对需要彻底删除的记录，必须先做逻辑删除，然后再做物理删除，逻辑删除是物理删除的前提条件。

逻辑删除数据表中的记录有两种方式。

#### (1) 菜单方式。

操作步骤 1。

- 打开表，进入表的浏览或编辑窗口。





- 在表的浏览或编辑窗口中，将鼠标移到要进行逻辑删除的记录上，对准其删除标志栏（每个记录的左前方有一个空白的小竖条），单击鼠标左键，这时要进行逻辑删除的记录就被加上了删除标记（左前方的空白竖条变黑），编辑模式下的情况如图 3-36 所示，浏览模式下的情况如图 3-37 所示。

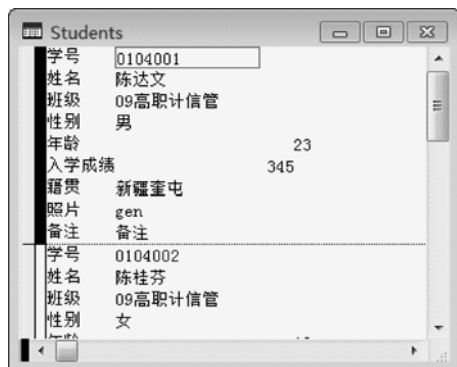


图 3-36 编辑模式下的情况



图 3-37 浏览模式下的情况

操作步骤 2。

- 打开表，进入表的浏览或编辑窗口。
- 在表的浏览或编辑窗口中，选择“表” “删除记录”命令，打开“删除”对话框，如图 3-38 所示。

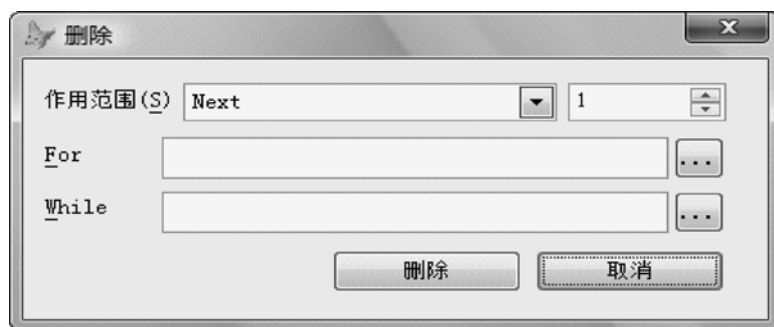


图 3-38 “删除”对话框

- 在“删除”对话框中，可以通过选择作用范围确定要进行逻辑删除的部分范围，也可以在 For 文本框中，设置逻辑删除条件表达式来确定逻辑删除的记录。
- 当确定了作用范围或条件表达式后，单击“删除”按钮，指定范围或满足条件的记录即被加上了删除标记。

(2) 命令方式。

命令格式：DELETE [<范围>] [FOR <条件>] [WHILE <条件>]

功能：对当前表文件中指定的记录做删除标记。

说明：用 DELETE 命令进行逻辑删除（加上删除标记），被逻辑删除的记录仍保留在表文件中，用 LIST 命令显示记录时，被逻辑删除的记录在记录号后面用“\*”号表示。

若执行了命令 SET DELETED ON，则操作时忽略被逻辑删除的记录；若执行了 SET DELETED OFF 命令，则操作时可以访问被逻辑删除的记录。

## 2. 恢复表中逻辑删除的记录

对已加上删除标记的记录，若用户想恢复其使用权，可对其进行恢复操作。恢复逻辑删除的记录，也有两种方式。

### (1) 菜单方式。

操作步骤 1。

- 打开表，进入表的浏览或编辑窗口。
- 在表的浏览或编辑窗口中，把鼠标移到要恢复的记录上，对准其删除标志栏，单击左键，去除删除标记。

操作步骤 2。

- 打开表，进入表的浏览或编辑窗口。
- 在表的浏览或编辑窗口中，选择“表” “恢复记录”命令，打开“恢复记录”对话框，如图 3-39 所示。

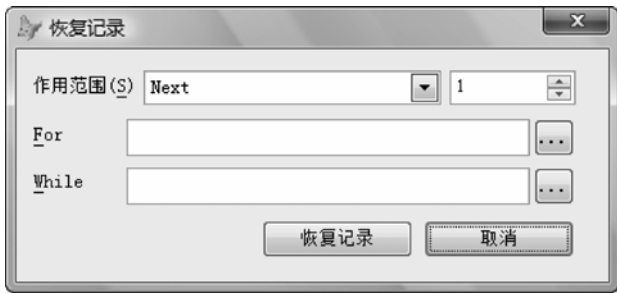


图 3-39 “恢复记录”对话框

- 在“恢复记录”对话框中，可以通过选择作用范围或条件表达式确定要进行恢复的部分记录。
- 当确定了作用范围或条件表达式后，单击“恢复记录”按钮，指定范围或满足条件的记录即被去除了删除标记。

### (2) 命令方式。

命令格式：RECALL [<范围>] [FOR <条件>] [WHILE <条件>]

功能：恢复当前表中带删除标记的记录，即去掉删除标记“\*”。

说明：当省略所有选项时，仅恢复当前记录。

## 3. 物理删除表中的记录

物理删除就是把无效的记录彻底从磁盘中删除。要想彻底删除数据表中无效的数据，必须先做逻辑删除，然后再做物理删除。物理删除表中的记录有以下两种方式。

### (1) 菜单方式。

操作步骤如下。

打开表，并进入表的浏览或编辑窗口。

若已对若干记录做了逻辑删除，则在表的浏览或编辑窗口中选择“表” “彻底删除”命令，弹出系统提示对话框，如图 3-40 所示。



单击“是”按钮，数据表中所有带有删除标记的记录将被彻底删除。



图 3-40 彻底删除系统提示对话框

(2) 命令方式。

命令格式：PACK

功能：在当前表中永久删除已被逻辑删除的记录。

#### 4. 删除所有记录

(1) 命令格式：ZAP

(2) 功能：从数据表中删除所有记录，只留下数据表的结构，即为空表。

(3) 说明：ZAP 命令与 DELETE ALL 和 PACK 两条命令的执行结果相同，区别在于 ZAP 的执行速度更快，这点在表中记录很多时尤为明显。

#### 5. 删除操作

对学生信息表 students 的记录进行以下删除操作。

(1) 逻辑删除第 2 个、第 4 个和第 8 个记录，再恢复这些逻辑删除的记录。

操作步骤如下。

菜单方式。

- 打开表，并进入表浏览窗口，如图 3-41 所示。
- 在表浏览窗口中，将鼠标分别移到第 2 个、第 4 个和第 8 个记录上，对准其删除标记栏，单击鼠标左键，这时要进行逻辑删除的记录就被加上了删除标记，如图 3-42 所示。

| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成 | 籍贯   | 照片  | 删除                                  |
|---------|-----|---------|----|----|-----|------|-----|-------------------------------------|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | 345 | 新疆奎屯 | gen | <input type="checkbox"/>            |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240 | 新疆伊犁 | gen | <input checked="" type="checkbox"/> |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420 | 广东高州 | gen | <input type="checkbox"/>            |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380 | 广东湛江 | gen | <input checked="" type="checkbox"/> |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290 | 广东徐闻 | gen | <input type="checkbox"/>            |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315 | 广东湛江 | gen | <input type="checkbox"/>            |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300 | 广东廉江 | gen | <input type="checkbox"/>            |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320 | 广东信宜 | gen | <input checked="" type="checkbox"/> |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310 | 广东揭阳 | gen | <input type="checkbox"/>            |

图 3-41 表浏览窗口

- 对准被加上删除标记记录的删除标记栏，单击左键，删除标记被清除，逻辑删除的记录就可以恢复。

| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成 | 籍贯   | 照片  |
|---------|-----|---------|----|----|-----|------|-----|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | 345 | 新疆奎屯 | gen |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240 | 新疆伊犁 | gen |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420 | 广东高州 | gen |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380 | 广东湛江 | gen |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290 | 广东徐闻 | gen |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315 | 广东湛江 | gen |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300 | 广东廉江 | gen |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320 | 广东信宜 | gen |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310 | 广东揭阳 | gen |

图 3-42 添加逻辑删除标记

命令方式。

```
USE F:\学生学籍管理系统\STUDENTS.DBF
DELETE RECORD 2
DELETE RECORD 4
DELETE RECORD 8
RECALL ALL
```

(2) 逻辑删除表中女生的记录，再恢复这些逻辑删除的记录。

操作步骤如下。

菜单方式。

- 打开表，并进入表浏览窗口，再选择“表” “删除记录”命令，打开“删除”对话框，如图 3-43 所示，在“作用范围”下拉列表框中选择 All，在条件 For 文本框中输入：Students.性别="女"。
- 单击“删除”按钮，表中的满足条件的记录即被加上了删除标记，如图 3-44 所示。
- 选择“表” “恢复记录”命令，打开“恢复记录”对话框，在“范围”下拉列表框中选择“ All ”，在 For 文本框中输入：Students.性别="女"，单击“恢复记录”按钮，表中的逻辑删除的记录标记即被去除，如图 3-45 所示。



图 3-43 “删除”对话框

命令方式。

```
DELETE ALL FOR Students.性别 = "女"
RECALL ALL
```

| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成 | 籍贯   | 照片  |
|---------|-----|---------|----|----|-----|------|-----|
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240 | 新疆伊犁 | gen |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420 | 广东高州 | gen |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380 | 广东湛江 | gen |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290 | 广东徐闻 | gen |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315 | 广东湛江 | gen |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300 | 广东廉江 | gen |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320 | 广东信宜 | gen |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310 | 广东揭阳 | gen |
|         |     |         |    | 0  |     |      | gen |

图 3-44 删除符合条件的记录

恢复记录

作用范围(S) All

For students. 性别="女"

While

恢复记录 取消

图 3-45 “恢复记录”对话框

(3) 逻辑删除表中班级是“09 高职多媒体”的同学的记录，学生毕业后，对这些记录进行物理删除。

菜单方式。

- 打开表，并进入表浏览窗口，再选择“表” “删除记录”命令，打开“删除”对话框，在“作用范围”下拉列表框中选择 All，在 For 文本框中输入：Students. 班级="09 高职多媒体"。
- 单击“删除”按钮，表中满足条件的记录即被加上了删除标记，如图 3-46 所示。

| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成 | 籍贯   | 照片  |
|---------|-----|---------|----|----|-----|------|-----|
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240 | 新疆伊犁 | gen |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420 | 广东高州 | gen |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380 | 广东湛江 | gen |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290 | 广东徐闻 | gen |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315 | 广东湛江 | gen |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300 | 广东廉江 | gen |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320 | 广东信宜 | gen |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310 | 广东揭阳 | gen |
|         |     |         |    | 0  |     |      | gen |

图 3-46 给记录添加删除标记

- 在表浏览窗口中选择“表” “彻底删除”命令，弹出系统提示对话框，如图 3-47 所示。
- 单击“是”按钮，表中所有逻辑删除的记录即被物理删除。通过浏览操作可以观察删除操作的结果。



图 3-47 系统提示对话框

命令方式。

```
DELETE ALL FOR Students.班别 = "09 高职多媒体"
PACK
```

### 3.2.9 记录的查询定位

在 Visual FoxPro 系统中，表中的每列数据都是通过字段名来标志的，每行数据都是通过记录号来标志的。用户通过记录号与字段名对表中每个数据项进行访问，指明了记录号（行）和字段名（列），即可确定表数据项。向表中输入数据时，系统按输入的顺序给每个记录一个记录号，最先输入的记录为 1 号记录，其次为 2 号记录，以此类推。

在 Visual FoxPro 系统内部，有一个确定当前记录的记录指针，记录指针指向的记录是当前记录。打开表文件时，记录指针总是指向第一个记录。表文件一旦打开，指针就可以移动，可见，当前记录是某个时刻的操作对象。

Visual FoxPro 系统移动记录指针的操作有菜单方式和命令方式两种。

#### 1. 菜单方式

操作步骤如下。

（1）打开要使用的表文件，进入该表的浏览窗口。

（2）在菜单栏上选择“表”|“转到记录”命令，弹出“转到记录”子菜单，从中选择不同的命令，可以将记录指针移动到相应的记录上。

若选择“第一个”命令，则定位第一个记录为当前记录。

若选择“最后一个”命令，则定位最后一个记录为当前记录。

若选择“下一个”命令，则定位当前记录的下一个记录为当前记录。

若选择“上一个”命令，则定位当前记录的前一个记录为当前记录。

若选择“记录号”命令，则弹出“转到记录”对话框，选择记录号，单击“确定”按钮，所选择的记录号即为当前记录。

若选择“定位”命令，则弹出“定位记录”对话框，如图 3-48 所示。输入定位条件表达式，即可确定满足条件表达式的第一条记录为当前记录。

#### 2. 命令方式

使用命令方式确定当前记录。

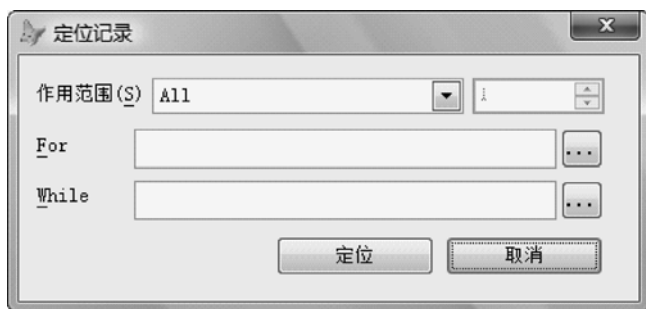


图 3-48 “定位记录”对话框

### (1) 绝对移动。

命令格式：GO/GOTO [RECORD] <记录号>

GO/GOTO TOP/BOTTOM

功能：将记录指针移动到指定记录上。

参数描述如下。

[RECORD]<记录号>指定一个物理记录号，该记录将成为当前记录。

可以省略命令动词 GO 或 GOTO 而只指定记录号。

该命令所指的物理记录号是该记录在表中的物理顺序号，执行命令格式 1 时，无论文件是否打开，均移动到物理记录号所指的记录，与表的逻辑顺序无关。

### (2) 相对移动。

命令格式：SKIP <+/-记录数>

功能：使记录指针在表中向前或向后相对移动。

参数描述如下。

短语<+/-记录数>用于指定记录指针需要移动的记录数。若命令短语中的记录数为正，则记录指针向文件尾方向移动指定的记录个数；若命令短语中的记录数为负，则记录指针向文件开头方向移动指定的记录个数。若命令中不带短语<+/-记录数>，则记录指针移动到下一个记录，相当于命令 SKIP 1。

### (3) 条件定位。

命令格式：LOCATE [<范围>] FOR <表达式>

功能：将记录指针指向第一个满足条件表达式的记录。

## 3. 操作案例

对学生信息表 students 的记录进行以下记录定位操作：确定第 5 条记录为当前记录；确定性别为女生的第一个记录为当前记录；确定当前记录的下一条记录为当前记录；确定最末一个记录为当前记录。

### (1) 菜单方式。

操作步骤如下。

打开表，进入表浏览窗口。

选择“表” “转到记录”命令，弹出“转到记录”子菜单。

在“转到记录”子菜单中选择“记录号”命令，打开“转到记录”对话框，如图 3-49 所示。



图 3-49 “转到记录”对话框

在“转到记录”对话框中输入或设置数值 5，再单击“确定”按钮。此时，5 号记录被定义为当前记录，如图 3-50 所示。

| Students |         |     |         |    |    |      |      |     |    |
|----------|---------|-----|---------|----|----|------|------|-----|----|
|          | 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  | 备注 |
|          | 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240  | 新疆伊犁 | gen |    |
|          | 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420  | 广东高州 | gen |    |
|          | 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380  | 广东湛江 | gen |    |
| ▶        | 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290  | 广东徐闻 | gen |    |
|          | 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315  | 广东湛江 | gen |    |
|          | 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300  | 广东廉江 | gen |    |
|          | 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 | gen |    |
|          | 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310  | 广东揭阳 | gen |    |
|          |         |     |         |    | 0  |      |      | gen |    |

图 3-50 定位到 5 号记录的结果

在“转到记录”子菜单中单击“定位”命令，打开“定位记录”对话框，作用范围选择“All”，在 For 文本框中输入定位条件：性别=“女”，如图 3-51 所示。

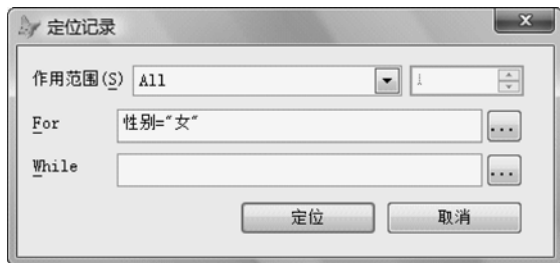


图 3-51 “定位记录”对话框

单击“定位”按钮，第一个满足定位条件的记录即被定义为当前记录（此时是第 2 条记录），如图 3-52 所示。

| Students |         |     |         |    |    |      |      |     |    |
|----------|---------|-----|---------|----|----|------|------|-----|----|
|          | 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  | 备注 |
|          | 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | 345  | 新疆奎屯 | gen |    |
| ▶        | 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240  | 新疆伊犁 | gen |    |
|          | 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420  | 广东高州 | gen |    |
|          | 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380  | 广东湛江 | gen |    |
|          | 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290  | 广东徐闻 | gen |    |
|          | 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315  | 广东湛江 | gen |    |
|          | 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300  | 广东廉江 | gen |    |
|          | 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 | gen |    |
|          | 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310  | 广东揭阳 | gen |    |

图 3-52 条件定位的结果





在“转到记录”子菜单中选择“下一个”命令，则第2条记录的下一条记录为当前记录，如图3-53所示。



| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  |
|---------|-----|---------|----|----|------|------|-----|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | 345  | 新疆奎屯 | gen |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240  | 新疆伊犁 | gen |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420  | 广东高州 | gen |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380  | 广东湛江 | gen |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290  | 广东徐闻 | gen |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315  | 广东湛江 | gen |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300  | 广东廉江 | gen |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 | gen |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310  | 广东揭阳 | gen |

图 3-53 转到下一个记录的结果

在“转到记录”子菜单中单击“最后一个”命令，则最后一个记录为当前记录，如图3-54所示。



| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  |
|---------|-----|---------|----|----|------|------|-----|
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315  | 广东湛江 | gen |
| 0104007 | 谭旭金 | 09高职多媒体 | 男  | 22 | 300  | 广东廉江 | gen |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 | gen |
| 0041002 | 李小华 | 09高职多媒体 | 女  | 23 | 310  | 广东揭阳 | gen |
| 0041003 | 张晶林 | 09高职计信管 | 男  | 22 | 310  | 河南郑州 | gen |

图 3-54 转到最后一个记录的结果

## (2) 命令方式。

在命令窗口中输入以下命令：

```
USE G:\MANAGER\STUDENTS.DBF
GOTO 5
LOCATE ALL FOR 性别="女"
SKIP
GO BOTTOM
```

### 3.2.10 在项目管理器中建立索引

#### 1. 利用表设计器的“字段”选项卡建立索引文件

这种方法只能建立普通索引。操作步骤如下。

- (1) 打开表，并进入“表设计器”对话框。
- (2) 在“表设计器”对话框中选择“字段”选项卡。
- (3) 选择要建立索引的关键字段。

(4) 在“索引”下拉列表框中选择“无”、“升序”或“降序”选项，建立索引。

(5) 单击“确定”按钮。

## 2. 利用表设计器的“索引”选项卡建立索引文件

操作步骤如下。

(1) 打开表，并进入“表设计器”对话框。

(2) 在“表设计器”对话框中选择“索引”选项卡，再根据需要选择参数。

在“排序”列选择索引方向，即升序（ ）或降序（ ）。

在“索引名”列输入索引字段名。

在“类型”列选择索引类型。

在“表达式”列输入索引字段名，或者索引表达式。

在“筛选”列限制记录的输出范围。

(3) 单击“确定”按钮。

## 3. 操作实例

利用表设计器为课程表 objects 中的“课程编号”字段按升序建立主索引，为“开设学期”字段按升序建立普通索引，为“课时”字段按降序建立普通索引。

操作步骤如下。

(1) 在项目管理器中，选择“数据”选项卡中的课程表 objects，并单击“修改”按钮，进入“表设计器”对话框。

(2) 建立第一索引。单击“课程编号”字段，选择索引为“升序”。

(3) 建立第二索引。单击“开设学期”字段，选择索引为“升序”。

(4) 建立第三索引。单击“课时”字段，选择索引为“降序”，如图 3-55 所示。

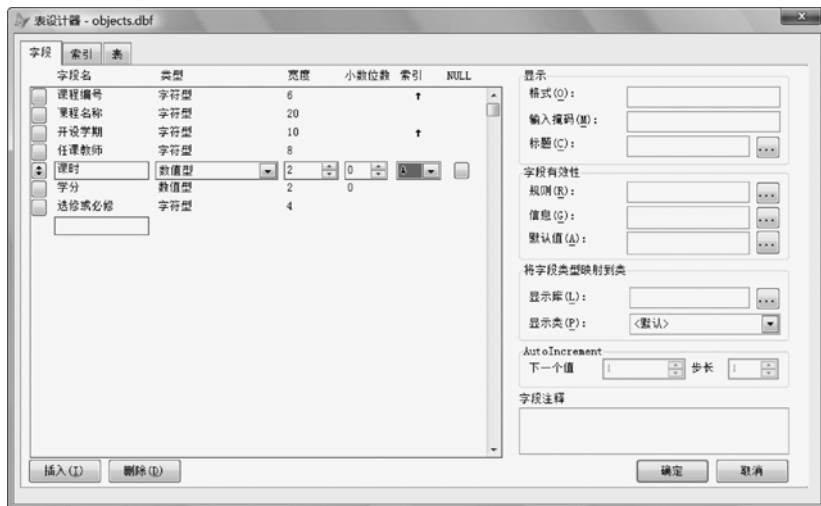


图 3-55 建立索引

(5) 单击“索引”标签，打开“索引”选项卡，在其中可以选择索引的类型、表达式和筛选等内容，其中类型有“主索引”、“候选索引”、“二进制”和“普通索引”4个选项，只需要单击“类型”下拉列表框即可选择索引类型，如图 3-56 所示。



图 3-56 选择索引类型

(6) 单击“确定”按钮，即可完成索引的设置。这时系统会自动比较目前的记录数据是否违反索引关键字设定的规则，弹出如图 3-57 所示的提示对话框，单击“是”按钮即可。

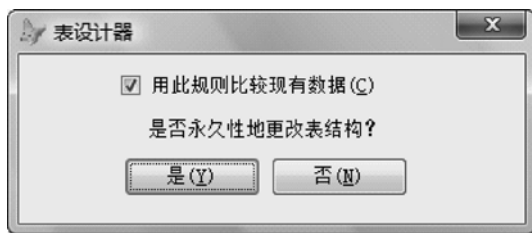


图 3-57 提示对话框

### 3.2.11 利用菜单打开索引

操作步骤如下。

(1) 在“项目管理器”中选择课程表 objects，单击“浏览”按钮，弹出数据表浏览窗口。

(2) 选择“表”“属性”命令，弹出如图 3-58 所示的“工作区属性”对话框。

(3) 在“索引顺序”下拉列表框中，系统默认“无顺序”选项，表示不按照任何索引来排序，只按数据在表中的物理位置顺序显示。

(4) 单击“索引顺序”下拉列表框，其中列出了数据表中所有索引项，索引列表如图 3-59 所示。

(5) 选择该下拉列表框中的某个索引项，单击“确定”按钮，即可按照以所选索引项为关键字的排列方式显示。假设选定了“课程编号”索引项，屏幕显示的结果如图 3-60 所示，该索引是以“课程编号”为关键字升序排列的。

(6) 若选择“课时”索引项，屏幕显示的结果如图 3-61 所示，该索引是以“课时”为关键字的降序排列。



图 3-58 “工作区属性”对话框

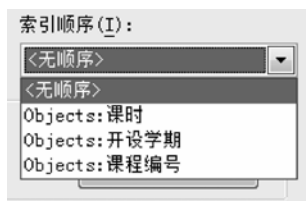


图 3-59 索引列表

| 课程编号   | 课程名称     | 开设学期 | 任课教师 |
|--------|----------|------|------|
| jsj001 | 操作系统     | 第一学期 | 赵玉   |
| jsj002 | VB语言     | 第一学期 | 钱红   |
| jsj003 | 计算机基础    | 第一学期 | 孙南   |
| jsj004 | 计算机网络    | 第二学期 | 郑东   |
| jsj005 | 面向对象技术   | 第三学期 | 李兰   |
| jsj006 | Delphi应用 | 第三学期 | 周五   |

图 3-60 以“课程编号”为关键字的升序排列

| 课程编号   | 课程名称   | 开设学期 | 任课教师 | 课时 | 学分 | 选修或必修 |
|--------|--------|------|------|----|----|-------|
| jsj001 | 操作系统   | 第一学期 | 赵玉   | 50 | 5  | 必修    |
| jsj005 | 面向对象技术 | 第三学期 | 李兰   | 45 | 4  | 必修    |
| jsj003 | 计算机基础  | 第一学期 | 孙南   | 45 | 2  | 必修    |
| jsj002 | VB语言   | 第一学期 | 钱红   | 45 | 4  | 必修    |
| jsj004 | 计算机网络  | 第二学期 | 郑东   | 40 | 3  | 选修    |

图 3-61 以“课时”为关键字的降序排列

### 3.2.12 对“学生信息表”进行排序

排序是将关键字段值相同的记录按顺序存放在一起，生成一个新的表文件。用于排序的字段称为关键字。

命令格式：SORT TO <表文件名> ON <字段名 1> [/A | /D] [/C] [, <字段名 2> [/A | /D] [/C] ...] [ASCENDING | DESCENDING] [<范围>] [FOR <逻辑表达式>] [WHILE <逻辑表达式>] [FIELDS <字段名列表>]

各参数的描述在上一个任务中已经介绍。

例如，对学生信息表 students 中的男生记录按“姓名”字段升序排列，只含有“姓名”、



“性别”和“年龄”字段，生成排序文件名为 std.dbf。

在命令窗口输入命令：

```
USE STUDENTS
SORT TO STD ON 姓名 FIELDS 姓名,性别,年龄 FOR 性别="男"
USE STD &&打开排序文件
LIST
```

屏幕显示结果如图 3-62 所示。

| 记录号 | 姓名  | 性别 | 年龄 |
|-----|-----|----|----|
| 1   | 陈达文 | 男  | 23 |
| 2   | 陈玉强 | 男  | 22 |
| 3   | 洪明信 | 男  | 23 |
| 4   | 江培强 | 男  | 22 |
| 5   | 林江文 | 男  | 22 |
| 6   | 林世鑫 | 男  | 26 |
| 7   | 谭旭金 | 男  | 22 |
| 8   | 张晶林 | 男  | 22 |

图 3-62 屏幕显示结果

### 3.2.13 设置域完整性约束

域约束规则在表设计器中建立。表设计器的“字段”选项卡中有一组定义字段有效性规则的项目，如图 3-63 所示，它们是“规则”(字段有效性规则)、“信息”(违背字段有效性规则时的提示信息)和“默认值”(字段的默认值)3 项。

|         |                      |     |
|---------|----------------------|-----|
| 字段有效性   |                      |     |
| 规则(R):  | <input type="text"/> | ... |
| 信息(I):  | <input type="text"/> | ... |
| 默认值(D): | <input type="text"/> | ... |

图 3-63 定义字段有效性规则的项目

操作步骤如下。

- (1) 选择表，在表设计器中打开。
- (2) 选择要定义字段有效性规则的字段。
- (3) 分别编辑规则、信息和默认值等项目。
- (4) 单击“确定”按钮。

字段有效性规则的项目可以直接输入，也可以单击右侧的按钮打开表达式生成器编辑、生成相应的表达式。

例如，在学生信息表 students 中，“性别”字段只能输入“男”或“女”，则在“规则”文本框中输入（或由表达式生成器生成）：性别="男".OR. 性别="女"

在“信息”文本框中输入：“性别只能为男或女”

在“默认值”文本框中输入：“男”

以上操作如图 3-64 所示。



图 3-64 字段有效性设置

如果输入的信息不能满足有效性规则，那么有效性说明信息会显示出来。当追加新记录时，如果“性别”输入了非法数据，那么会弹出如图 3-65 所示的系统提示对话框。



图 3-65 系统提示对话框

这正是在“信息”文本框中输入的说明。

另外，在建立有效性规则时，可能要考虑这样一些问题：字段的长度、字段可能为空或包含了已设置好的值等。

例如，假设现在正使用学生信息表，想确保新记录的“籍贯”字段的内容少于 13 个字符，则可在“籍贯”字段的“规则”文本框中输入：

```
LEN(ALLTRIM(籍贯)) < 13
```

然后在“信息”文本框中输入下述错误提示：“籍贯字段内容不能超过 13 个字符，请重新输入”。如果输入的籍贯太长，就会弹出一个对话框，其中会显示有效性说明。

注意：规则是逻辑表达式，信息是字符串表达式，“默认值”的类型则由字段的类型确定。此外，以上域约束规则是对数据库表而言的，自由表不能建立字段级规则和约束。



### 3.2.14 建立各表之间的关系

利用数据库设计器可以直接、方便地为数据库表建立关系，并且建立的关系作为数据库的一部分被保存起来。每当用户在查询设计器、视图设计器或数据环境设计器中使用表时，这些关系将作为表之间的默认连接。

例如，在数据库设计器中创建关系。

操作步骤如下。

(1) “学籍管理数据库”中有3个表，其中学生信息表 students 以“学号”字段建立主索引，课程表 objects 以“课程编号”建立主索引，成绩表 scores 以“学号”、“课程编号”建立普通索引。

(2) 在数据库设计器中打开“学籍管理数据库”。

(3) 将学生信息表 students 的主索引“学号”拖动到成绩表 scores 的索引“学号”上，将课程表 objects 中主索引“课程编号”拖动到成绩表 scores 的索引“课程编号”上，两表之间会出现一条表示关系的连线，如图 3-66 所示。

从图 3-66 中可以看出，学生信息表 students 与成绩表 scores、课程表 objects 与成绩表 scores 之间的关系是一对多的关系。

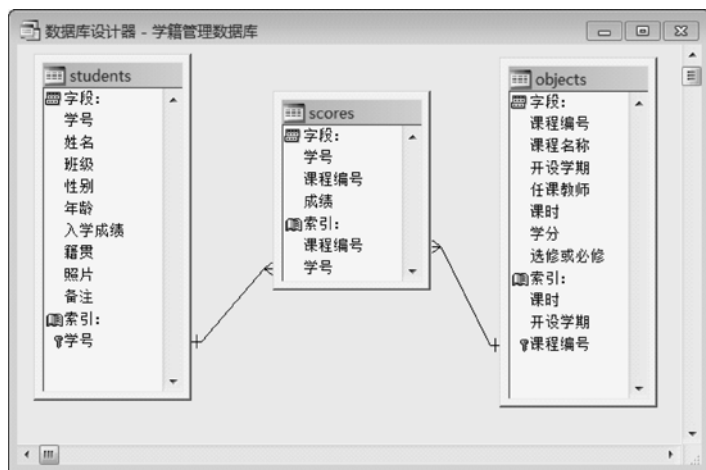


图 3-66 为“学籍管理数据库”中的表建立关系

**注意：**在建立永久关系之前，必须先为主表和子表以相同的关键字建立索引，并且主表中的索引必须为主索引或候选索引。

右击两表间的关系连线，并从弹出的快捷菜单中选择“删除关系”选项，可以删除选定的关系，两表间的关系连线同时消失；也可以单击连线后，按 Delete 键来删除选定的关系。

右击两表间的关系连线，并从弹出的快捷菜单中选择“编辑关系”选项，可以打开“编辑关系”对话框编辑选定的关系。

### 3.2.15 设置参照完整性约束

建立关系后,就可以设置、管理数据库的参照完整性了。“参照完整性生成器”可以帮助我们建立规则,控制记录在相关表中的插入、更新或删除等操作,这些规则将被写到相应的表触发器中。

操作步骤如下。

(1) 在数据库设计器中打开“学籍管理数据库”。

(2) 选择“数据库”→“编辑参照完整性”命令,打开“Referential Integrity Builder (参照完整性生成器)”对话框,如图 3-67 所示。

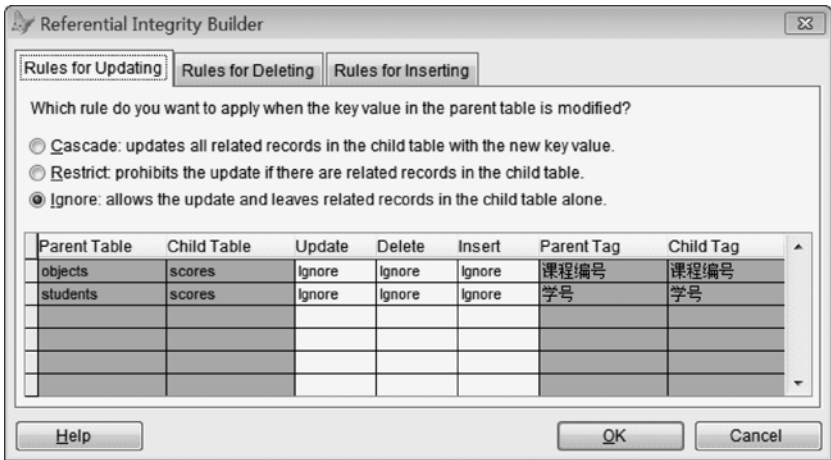


图 3-67 “Referential Integrity Builder (参照完整性生成器)”对话框

(3) 选择“Rules for Updating (更新规则)”选项卡,在其中可以利用 3 个单选按钮设置关联表间的更新规则。

若选中“Cascade (级联)”单选按钮,则用新的关键字值更新子表中的所有相关记录。

选中“Restrict (限制)”单选按钮,若子表中有相关记录则禁止更新。

选中“Ignore (忽略)”单选按钮,允许更新,但不更新子表中的相关记录,使其成为孤记录。

(4) 选择“Rules for Deleting (删除规则)”选项卡,可以利用 3 个单选按钮设置关联表间的删除规则。

若选中“Cascade (级联)”单选按钮,则删除子表中的所有相关记录。

选中“Restrict (限制)”单选按钮,若子表中有相关记录则禁止删除。

选中“Ignore (忽略)”单选按钮,允许删除,但不管子表中的相关记录。

(5) 选择“Rules for Inserting (插入规则)”选项卡,可以利用两个单选按钮设置关联表间的插入规则。

选中“Restrict (限制)”单选按钮,若父表中没有匹配的关键字值,则禁止插入。

选中“Ignore (忽略)”单选按钮,允许插入。

(6) 单击“确定”按钮,然后单击“是”按钮保存所做的修改,生成参照完整性代码,





并退出参照完整性生成器。这样，参照完整性就可利用两表的关系参照制约来控制两表数据的完整性和一致性。

### 任务 3.3 项目分析

在这个项目中，我们为学生学籍管理系统建立了 3 个主要的表：学生信息表 students，课程表 objects，成绩表 scores。同时，我们学习了如何对单独的表或几个相关的表进行操作，下面我们对该项目中的表进行逐一分析。

#### 3.3.1 学生信息表

学生信息表如表 3-2 所示。

表 3-2 学生信息表

| 字段名称 | 字段类型 | 字段长度 | 小数位数 |
|------|------|------|------|
| 学号   | 字符型  | 8    |      |
| 姓名   | 字符型  | 10   |      |
| 班级   | 字符型  | 12   |      |
| 性别   | 字符型  | 2    |      |
| 年龄   | 整型   | 4    |      |
| 入学成绩 | 数值型  | 10   | 0    |
| 籍贯   | 字符型  | 10   |      |
| 照片   | 通用型  | 4    |      |
| 备注   | 备注型  | 4    |      |

其中以“学号”建立主索引。

#### 3.3.2 课程表

课程表如表 3-3 所示。

表 3-3 课程表

| 字段名称  | 字段类型 | 字段长度 | 小数位数 |
|-------|------|------|------|
| 课程编号  | 字符型  | 6    |      |
| 课程名称  | 字符型  | 20   |      |
| 开设学期  | 字符型  | 10   |      |
| 任课教师  | 字符型  | 8    |      |
| 课时    | 数值型  | 2    | 0    |
| 学分    | 数值型  | 2    | 0    |
| 选修或必修 | 字符型  | 4    |      |

其中以“课程编号”建立主索引。

#### 3.3.3 成绩表

成绩表如表 3-4 所示。

表 3-4 成绩表

| 字段名称 | 字段类型 | 字段长度 | 小数位数 |
|------|------|------|------|
| 学号   | 字符型  | 8    |      |
| 课程编号 | 字符型  | 7    |      |
| 成绩   | 数值型  | 10   | 0    |

其中以“学号”和“课程编号”建立普通索引。

## 任务 3.4 模仿设计与自主设计

### 3.4.1 创建表

(1) 为“图书管理系统”建立自由表“图书信息表”，如表 3-5 所示。

表 3-5 图书信息表

| 字段名称 | 字段类型 | 字段长度 | 小数位数 |
|------|------|------|------|
| 图书编号 | 字符型  | 8    |      |
| 图书名称 | 字符型  | 20   |      |
| 作者   | 字符型  | 10   |      |
| 出版社  | 字符型  | 20   |      |
| 出版日期 | 日期型  | 8    |      |
| 备注   | 备注型  | 4    |      |
| 价格   | 货币型  | 8    |      |
| 数量   | 数值型  | 4    | 0    |

(2) 在“图书管理”数据库中创建数据库表“借阅者信息表”，如表 3-6 所示。

表 3-6 借阅者信息表

| 字段名称 | 字段类型 | 字段长度 | 小数位数 |
|------|------|------|------|
| 借书证号 | 字符型  | 8    |      |
| 姓名   | 字符型  | 10   |      |
| 性别   | 字符型  | 2    |      |
| 身份证  | 字符型  | 18   |      |
| 联系电话 | 字符型  | 12   |      |
| 密码   | 字符型  | 20   |      |

(3) 在“图书管理”数据库中创建数据库表“租借信息表”，如表 3-7 所示。

表 3-7 租借信息表

| 字段名称 | 字段类型  | 字段长度 | 小数位数 |
|------|-------|------|------|
| 借书证号 | 字符型   | 8    |      |
| 图书编号 | 字符型   | 8    |      |
| 借阅日期 | 日期时间型 | 8    |      |
| 是否归还 | 字符型   | 2    |      |
| 归还日期 | 日期时间型 | 8    |      |

(4) 将图书信息表添加到“图书管理”数据库中。



### 3.4.2 表的基本操作

- (1) 利用菜单和命令两种方式打开上述已经创建的表。
- (2) 利用菜单方式修改“图书信息表”。
- (3) 利用命令方式修改“借阅者信息表”。
- (4) 在项目管理器中修改“租借信息表”。
- (5) 关闭上述所有的表。

### 3.4.3 表中数据的操作

- (1) 分别向上面建立的3个表中输入数据。
- (2) 在“图书信息表”中追加新进的图书的数据。
- (3) 用项目管理器、菜单和命令3种方式浏览上述3个表中的数据。
- (4) 用项目管理器、菜单和命令3种方式修改上述3个表中有误的数据。

### 3.4.4 创建数据库中多表之间的联系

- (1) “图书信息表”以“图书编号”为关键字升序建立主索引。
- (2) “借阅者信息表”以“借书证号”为关键字升序建立主索引。
- (3) “租借信息表”以“图书编号”为关键字升序排列建立普通索引，以“借书证号”为关键字升序排列建立普通索引。
- (4) 建立各表之间的联系。  
图书信息表．图书编号=租借信息表．图书编号  
借阅者信息表．借书证号=租借信息表．借书证号

### 3.4.5 表中字段属性设置

- (1) 设置“借阅者信息表”中“性别”的字段有效性。  
规则：性别="男".OR. 性别="女"  
信息："性别只能为男或女"  
默认值："男"
- (2) 设置“借阅者信息表”中“身份证”的字段有效性。  
规则：LEN(ALLTRIM(身份证))=18  
信息："身份证字段内容只能是18个字符，请重新输入"

### 3.4.6 记录的定位和筛选

- (1) 筛选出“图书信息表”中“出版日期”为近一年的图书信息。
- (2) 筛选出尚未归还图书的读者的信息。

(3) 打开“图书信息表”，移动记录到底部，再移动记录到顶部，然后移动记录到第 5 条，最后移动到下一条记录。

(4) 在“图书信息表”中查找出“数量”为“0”的图书信息。

### 项目小结 3

项目 3 是这本教材的重点内容，详细介绍了表的建立和基本操作，索引的建立和使用方法，如何将表组织为一个数据库，表之间联系的表示，以及实体完整性、域完整性和参照完整性等内容。表是数据库的最主要组成元素，因此本项目介绍的内容是建立一个完整的系统所必备的基础性知识，在本书中占有重要的位置。

### 项目习题 3

#### 1. 选择题

- (1) 在 Visual FoxPro 中，调用表设计器建立数据库表 student.dbf 的命令是 ( )。
- A. MODIFY STRUCTURE      B. MODIFY COMMAND  
C. CREATE STUDENT      D. CREATE TABLE STUDENT
- (2) 在 Visual FoxPro 中，关于自由表叙述正确的是 ( )。
- A. 自由表和数据库表是完全相同的  
B. 自由表不能建立字段级规则和约束  
C. 自由表不能建立候选索引  
D. 自由表不可以加入到数据库中
- (3) 在 Visual FoxPro 中建立数据库表时，将年龄字段值限制在 12 ~ 14 的约束属于 ( )。
- A. 实体完整性约束      B. 域完整性约束  
C. 参照完整性约束      D. 视图完整性约束
- (4) 在创建数据库表结构时，给该表指定了主索引，这属于数据完整性中的 ( )。
- A. 参照完整性      B. 实体完整性  
C. 域完整性      D. 用户定义完整性
- (5) 在 Visual FoxPro 中，学生表 student 包含通用型字段，通用型字段中的数据都存储在 ( ) 文件中。
- A. student.doc      B. student.men  
C. student.dbt      D. student.ftp
- (6) 可以在表设计器的“字段”选项卡中创建的索引是 ( )。
- A. 唯一索引      B. 候选索引      C. 主索引      D. 普通索引



(7) 设有两个数据库表, 父表和子表间存在一对多联系, 为了控制子表和父表的关联, 可以设置参照完整性规则, 为此要求这两张表 ( )。

- A. 在父表连接字段上建立普通索引, 在子表连接字段上建立主索引
- B. 在父表连接字段上建立主索引, 在子表连接字段上建立普通索引
- C. 父表连接字段上不需要建立任何索引, 在子表连接字段上建立普通索引
- D. 父表和子表的连接字段上都要建立主索引

(8) 在指定字段或表达式中不允许出现重复值的索引是 ( )。

- A. 唯一索引
- B. 唯一索引和候选索引
- C. 唯一索引和主索引
- D. 主索引和候选索引

(9) 在 Visual FoxPro 中, 使用 LOCATE FOR <expL>命令按条件查找记录, 当查到满足条件的第 1 条记录后, 如果还需要查找下一条满足条件的记录, 应该使用 ( )。

- A. LOCATE FOR <expL>命令
- B. SKIP 命令
- C. CONTINUE 命令
- D. GO 命令

(10) 在 Visual FoxPro 中, 如果在表之间的联系中设置了参照完整性规则, 并在删除规则中选择了“限制”, 那么当删除父表中的记录时, 系统的反应是 ( )。

- A. 不做参照完整性检查
- B. 不准删除父表中的记录
- C. 自动删除子表中的所有相关记录
- D. 若子表中有相关记录, 则禁止删除父表中的记录

(11) 下列命令中, 不能对记录进行编辑修改的是 ( )。

- A. MODI STRU
- B. EDIT
- C. CHANGE
- D. BROWSE

(12) 在 Visual FoxPro 中, 逻辑删除表中性别为“女”的命令是 ( )。

- A. DELETE FOR 性别=女
- B. DELETE 性别=女
- C. PACK 性别=女
- D. ZAP 性别=女

(13) 打开数据库文件后, 当前记录指针指向 100, 要使指针指向记录号为 20 的记录, 应使用命令 ( )。

- A. LOCATE 20
- B. SKIP -80
- C. GO 20
- D. SKIP 80

(14) 要为当前表中所有学生的年龄增加 2 岁, 应输入的命令是 ( )。

- A. CHANGE ALL 年龄 WITH 年龄+2
- B. CHANGE ALL 年龄+2WITH 年龄
- C. REPLACE ALL 年龄+2WITH 年龄
- D. REPLACE ALL 年龄 WITH 年龄+2

(15) 当前数据表中共有 10 条记录, 且无索引文件处于打开状态, 若执行命令 GO 5 后接着执行命令 INSERT BLANK BEFORE, 则此时记录指针指向第 ( ) 条记录。

- A. 4
- B. 5
- C. 6
- D. 11

## 2. 填空题

- (1) 当删除父表中的记录时, 若子表中的所有相关记录也能自动删除, 则相应的参照完整性删除规则为\_\_\_\_\_。
- (2) 实现表之间临时联系的命令是\_\_\_\_\_。
- (3) 在定义字段有效性规则时, 在“规则”文本框中书写的表达式值的类型是\_\_\_\_\_。
- (4) 在 Visual FoxPro 中, 最多允许同时打开\_\_\_\_\_个表。
- (5) 使数据库表变为自由表的命令是\_\_\_\_\_ TABLE。
- (6) 把当前表当前记录的“学号, 姓名”字段值复制到数组 A 的命令是: SCATTER  
FIELD 学号, 姓名\_\_\_\_\_。



# 项目 4

## 设计学生学籍管理系统 中的查询与视图

### 项目要点

- 查询设计器的使用。
- 视图设计器的使用。

### 学习目标

- 掌握查询设计器的使用及建立查询的方法。
- 掌握视图设计器的使用。
- 理解掌握 SQL 语句的使用。

建立数据库并存储大量数据的最终目的，是使用户可以通过查询获得最关心、最想要的信息。数据检索是人们使用数据库时最频繁的操作之一，而查询和视图都是为快速、方便地使用数据库数据所采取的方法。

### 任务 4.1 熟悉查询与视图基本知识

#### 4.1.1 查询

查询是数据库系统中最常用的操作之一，查询效率的高低在很大程度上影响着程序的效率。查询及其优化是 Visual FoxPro 的一项关键技术。为了提供更快速、更强大的检索功

能，VFP 9.0 引入了 SQL ( Structured Query Language ) 结构化查询语言。用户可以用查询设计器建立查询文件，查询文件的扩展名是 .qpr，也可以用 SQL 语句来实现查询。

#### 4.1.2 视图

视图是一种比查询更高级的检索方式。它与查询有着本质的区别，视图可以更新数据，而查询不能更新数据，查询保存在一个独立的文件中，而视图不是独立的文件，它只能存储在数据库中。

视图是从一个或多个数据表中导出的“表”。它与数据表不同的是，视图中的数据还是存储在原来的数据表中，因此可以把它看做是一个“虚表”。视图不能单独存在，它依赖于某个数据库而存在，只有打开与视图相关的数据库才能创建和使用视图。

根据数据库中数据来源的不同，可以把视图分为本地视图和远程视图。

#### 4.1.3 SQL 概述

SQL 是结构化数据查询语言 ( Structured Query Language ) 的英文缩写，是一种使用关系模型的数据库应用语言，现在所有的关系数据库管理系统都支持 SQL。FoxPro 从 2.5 For DOS 版本就开始支持 SQL。

SQL 最早是在 20 世纪 70 年代由 IBM 公司开发出来的，并被应用在 DB2 关系数据库系统中，主要用于关系数据库中的信息检索。20 世纪 80 年代初，美国国家标准协会 ( ANSI ) 即着手开展 SQL 的标准化工作，1986 年 ANSI 的数据库委员会批准了 SQL 作为关系数据库语言的美国标准，即第一个 SQL 标准。后来，又在 1989 年和 1992 年相继对其进行了扩充和完善，即 ANSI SQL 89 和 ANSI SQL 92。这些标准的出台使 SQL 作为标准关系数据库语言的地位更加巩固，大多数数据库供应商纷纷采用 SQL 作为其产品的检索语言。

SQL 语言主要有以下四大功能。

- (1) 数据查询 ( DQL )：查询数据。
- (2) 数据定义 ( DDL )：建立、删除和修改数据对象。
- (3) 数据操纵 ( DML )：完成数据操作的命令，包括查询。
- (4) 数据控制 ( DCL )：控制对数据库的访问，服务器的关闭、启动等。

SQL 语言具有一体化、统一的语法结构和高度非过程化的特点，简单易学，风格统一，利用简单的几个英语单词的组合就可以完成所有的功能。

#### 4.1.4 SQL 查询功能

SQL 的核心是查询，SQL 的查询命令即 SELECT 命令，它的基本形式由 SELECT-FROM-WHERE 查询块组成，查询可以嵌套。

查询语句的格式为

```
SELECT [ALL/DISTINCT] */目标列
FROM 基表 (或视图)
[WHERE 条件表达式]
[GROUP BY 列名 1[HAVING 内部函数表达式]]
[ORDER BY 列名 2 ASC/DESC];
```





该语句的功能是根据 WHERE 子句中的条件表达式,从 FROM 子句给出的基表或视图中找到满足条件的元组,按 SELECT 子句中的目标列,选出元组中的分量形成结果表。GROUP 子句将按列名 1 分组,每个组产生的结果表示一个元组,有 HAVING 子句时,列名 1 按 HAVING 后的条件分组。如果有 ORDER 子句,则结果表要根据指定的列名 2 按升序或降序排序。

在上述格式中,SELECT 后是查询目标表,其中参数 ALL 检索符合条件的所有元组;DISTINCT 检索去掉重复的所有元组,默认值为 ALL;“\*”检索结果为整个元组,即包括所有的列。

我们以下面新给出的“学生成绩管理”数据库中的表为例,来学习 SQL 结构化查询语言,为了方便读者对照和验证查询结果,下面给出学生表、课程表和成绩表的具体记录值,依次如表 4-1,表 4-2 和表 4-3 所示。

表 4-1 学生表

学生(学号 C7,姓名 C8,年龄 I):

| 学 号  | 姓 名 | 年 龄 |
|------|-----|-----|
| 1001 | 张 明 | 18  |
| 1002 | 李晓红 | 16  |
| 1003 | 黄定光 | 21  |
| 1004 | 刘 庆 | 18  |
| 1005 | 张晓霞 | 22  |
| 1006 | 于安虎 | 18  |
| 1007 | 杜振宇 | 21  |
| 1008 | 安 宏 | 16  |

表 4-2 课程表

课程(课程号 C6,课程名 C14):

| 课 程 号 | 课 程 名 |
|-------|-------|
| KC01  | 语文    |
| KC02  | 高等数学  |
| KC03  | 程序设计  |
| KC04  | 大学英语  |
| KC05  | 人工智能  |

表 4-3 成绩表

成绩(学号 C7,课程号 C14,成绩 I):

| 学 号  | 课 程 号 | 成 绩 |
|------|-------|-----|
| 1001 | KC01  | 98  |
| 1001 | KC02  | 89  |
| 1001 | KC05  | 99  |
| 1002 | KC02  | 76  |
| 1002 | KC03  | 87  |
| 1003 | KC02  | 69  |
| 1003 | KC05  | 93  |
| 1004 | KC03  | 56  |
| 1004 | KC05  | 68  |
| 1005 | KC01  | 84  |
| 1006 | KC02  | 63  |
| 1007 | KC02  | 88  |

## 1. 简单查询

首先从几个最简单的查询开始，这些查询基于单个表，可以有简单的查询条件。这样的查询由 SELECT 和 FROM 子句构成无条件查询，或者由 SELECT、FROM 和 WHERE 子句构成条件查询。下面将举例说明。

【例 4-1】从成绩表中检索“课程号”字段的值。

在命令窗口输入命令：

```
SELECT 课程号 FROM 成绩
```

主屏幕显示的内容为：

```
课程号
KC01
KC02
KC05
KC02
KC03
KC02
KC05
KC03
KC05
KC01
KC02
KC02
```

在这里可以看到在结果中有重复值，如果要去掉重复值，只需要指定参数 DISTINCT。在命令窗口输入命令：

```
SELECT DISTINCT 课程号 FROM 成绩
```

主屏幕显示的内容为：

```
课程号
KC01
KC02
KC03
KC05
```

【例 4-2】从学生表中检索年龄大于 20 的所有元组。

在命令窗口输入命令：

```
SELECT * FROM 学生 WHERE 年龄>20
```

主屏幕显示的内容为：

| 学号   | 姓名  | 年龄 |
|------|-----|----|
| 1003 | 黄定光 | 21 |
| 1005 | 张晓霞 | 22 |
| 1007 | 杜振宇 | 21 |



在这里“\*”是通配符，表示所有的属性，WHERE 短语指定了查询条件，查询条件可以是任意复杂的逻辑表达式。

## 2. 连接查询

连接是关系的基本操作之一，连接查询是一种基于多个关系表的查询。在旧版的 FoxPro 中，一般使用下面的形式。

```
SELECT [表.]输出项 [AS 列名][[,表]输出项[AS 列名]...];
```

```
FROM 表[局部别名][,表[局部别名]...]WHERE 连接条件表达式[AND 连接条件表达式][AND | OR 筛选条件表达式]
```

【例 4-3】显示每个学生的姓名、课程名和成绩。

这里要求所检索的信息分别出自学生表、课程表和成绩表，这样的检索肯定是基于多个关系的。

在命令窗口输入命令：

```
SELECT 姓名,课程名,成绩 FROM 学生,课程,成绩;
WHERE 学生.学号=成绩.学号 AND 成绩.课程号=课程.课程号
```

主屏幕显示的内容为：

| 姓名  | 课程名  | 成绩 |
|-----|------|----|
| 张 明 | 语文   | 98 |
| 张 明 | 高等数学 | 89 |
| 张 明 | 人工智能 | 99 |
| 李晓红 | 高等数学 | 76 |
| 李晓红 | 程序设计 | 87 |
| 黄定光 | 高等数学 | 69 |
| 黄定光 | 人工智能 | 93 |
| 刘 庆 | 程序设计 | 56 |
| 刘 庆 | 人工智能 | 68 |
| 张晓霞 | 语文   | 84 |
| 于安虎 | 高等数学 | 63 |
| 杜振宇 | 高等数学 | 88 |

这里的“学生.学号=成绩.学号”、“成绩.课程号=课程.课程号”是两个连接条件。如果在 FROM 之后出现了两个以上的关系，那么这些关系之间肯定存在一种联系，否则无法构成检索表达式。此外，这种旧版的查询格式不考虑 FROM 之后的表名顺序，没有父表、子表之分；默认使用内连接的方法，即只显示满足连接条件的记录。

在 VFP 9.0 中可以使用下面的格式进入多表查询，格式如下。

```
SELECT...;
FROM Table INNER | LEFT | RIGHT | FULL JOIN Table;
ON JoinCondition;
WHERE...
```

其中：

(1) INNER JOIN 等价于 JOIN，为普通连接，在 Visual FoxPro 中称为内部连接，只显示父表与子表相匹配的记录；

- (2) LEFT JOIN 为左连接,即子表中没有匹配的记录也将显示父表中所有的记录;
- (3) RIGHT JOIN 为右连接,即父表中没有匹配的记录也将显示子表中所有的记录;
- (4) FULL JOIN 为全连接,即显示父表和子表所有的记录;
- (5) ON JoinCondition 指定连接条件。

从以上格式可以看出,它的连接条件在 ON 子句中给出,而不在 WHERE 子句中,连接类型在 FROM 子句中给出。

【例 4-4】内部连接,即只显示父表与子表相匹配的记录。

在命令窗口输入命令:

```
SELECT 学生.学号,姓名,课程号,成绩;
FROM 学生 INNER JOIN 成绩;
ON 学生.学号=成绩.学号
```

主屏幕显示的内容为:

| 学号   | 姓名  | 课程号  | 成绩 |
|------|-----|------|----|
| 1001 | 张明  | KC01 | 98 |
| 1001 | 张明  | KC02 | 89 |
| 1001 | 张明  | KC05 | 99 |
| 1002 | 李晓红 | KC02 | 76 |
| 1002 | 李晓红 | KC03 | 87 |
| 1003 | 黄定光 | KC02 | 69 |
| 1003 | 黄定光 | KC05 | 93 |
| 1004 | 刘庆  | KC03 | 56 |
| 1004 | 刘庆  | KC05 | 68 |
| 1005 | 张晓霞 | KC01 | 84 |
| 1006 | 于安虎 | KC02 | 63 |
| 1007 | 杜振宇 | KC02 | 88 |

【例 4-5】左连接,即除了满足连接条件的记录出现在查询结果中外,第一个表中没有匹配的记录也将显示在查询结果中。

在命令窗口输入命令:

```
SELECT 学生.学号,姓名,课程号,成绩;
FROM 学生 LEFT JOIN 成绩;
ON 学生.学号=成绩.学号
```

主屏幕显示的内容为:

| 学号   | 姓名  | 课程号  | 成绩 |
|------|-----|------|----|
| 1001 | 张明  | KC01 | 98 |
| 1001 | 张明  | KC02 | 89 |
| 1001 | 张明  | KC05 | 99 |
| 1002 | 李晓红 | KC02 | 76 |
| 1002 | 李晓红 | KC03 | 87 |
| 1003 | 黄定光 | KC02 | 69 |
| 1003 | 黄定光 | KC05 | 93 |
| 1004 | 刘庆  | KC03 | 56 |



|      |     |        |        |
|------|-----|--------|--------|
| 1004 | 刘庆  | KC05   | 68     |
| 1005 | 张晓霞 | KC01   | 84     |
| 1006 | 于安虎 | KC02   | 63     |
| 1007 | 杜振宇 | KC02   | 88     |
| 1008 | 安宏  | .NULL. | .NULL. |

为了看到右连接和全连接的效果，假设在成绩表中插入以下记录：

|        |        |    |
|--------|--------|----|
| "1009" | "KC03" | 67 |
|--------|--------|----|

【例 4-6】右连接，即除了满足连接条件的记录出现在查询结果中外，第二个表中没有匹配的记录也将显示在查询结果中。

在命令窗口输入命令：

```
SELECT 学生.学号,姓名,课程号,成绩;
FROM 学生 RIGHT JOIN 成绩;
ON 学生.学号=成绩.学号
```

主屏幕显示的内容为：

| 学号     | 姓名     | 课程号  | 成绩 |
|--------|--------|------|----|
| 1001   | 张明     | KC01 | 98 |
| 1001   | 张明     | KC02 | 89 |
| 1001   | 张明     | KC05 | 99 |
| 1002   | 李晓红    | KC02 | 76 |
| 1002   | 李晓红    | KC03 | 87 |
| 1003   | 黄定光    | KC02 | 69 |
| 1003   | 黄定光    | KC05 | 93 |
| 1004   | 刘庆     | KC03 | 56 |
| 1004   | 刘庆     | KC05 | 68 |
| 1005   | 张晓霞    | KC01 | 84 |
| 1006   | 于安虎    | KC02 | 63 |
| 1007   | 杜振宇    | KC02 | 88 |
| .NULL. | .NULL. | KC03 | 67 |

【例 4-7】全连接，即除了满足连接条件的记录出现在查询结果中外，两个表中不满足连接条件的记录也出现在查询结果中。

在命令窗口输入命令：

```
SELECT 学生.学号,姓名,课程号,成绩;
FROM 学生 FULL JOIN 成绩;
ON 学生.学号=成绩.学号
```

主屏幕显示的内容为：

| 学号   | 姓名  | 课程号  | 成绩 |
|------|-----|------|----|
| 1001 | 张明  | KC01 | 98 |
| 1001 | 张明  | KC02 | 89 |
| 1001 | 张明  | KC05 | 99 |
| 1002 | 李晓红 | KC02 | 76 |
| 1002 | 李晓红 | KC03 | 87 |

|        |        |        |        |
|--------|--------|--------|--------|
| 1003   | 黄定光    | KC02   | 69     |
| 1003   | 黄定光    | KC05   | 93     |
| 1004   | 刘庆     | KC03   | 56     |
| 1004   | 刘庆     | KC05   | 68     |
| 1005   | 张晓霞    | KC01   | 84     |
| 1006   | 于安虎    | KC02   | 63     |
| 1007   | 杜振宇    | KC02   | 88     |
| 1008   | 安宏     | .NULL. | .NULL. |
| .NULL. | .NULL. | KC03   | 67     |

### 3. 嵌套查询

嵌套查询也称为子查询，是指一个查询块 (SELECT-FROM-WHERE) 可以嵌入另一个查询块中。有时候用户需要根据已查询的结果来确定查询的筛选条件，也就是说甲查询的输出结果依赖于乙查询的输出，此时我们就称乙查询为甲查询的子查询。SQL 语言允许多层查询嵌套，并且内嵌的子查询要用圆括号括起来。

【例 4-8】查询与学号为 1001 的学生选修了同一门课程的学生的成绩。

在命令窗口输入命令：

```
SELECT * FROM 成绩 WHERE 课程号 IN;
(SELECT 课程号 FROM 成绩 WHERE 学号='1001')
```

主屏幕显示的内容为：

| 学号   | 课程号  | 成绩 |
|------|------|----|
| 1001 | KC01 | 98 |
| 1001 | KC02 | 89 |
| 1001 | KC05 | 99 |
| 1002 | KC02 | 76 |
| 1003 | KC02 | 69 |
| 1003 | KC05 | 93 |
| 1004 | KC05 | 68 |
| 1005 | KC01 | 84 |
| 1006 | KC02 | 63 |
| 1007 | KC02 | 88 |

这里的 IN 相当于集合运算符“ ”。

【例 4-9】查询所有课程成绩都不低于 80 的学生信息。

在命令窗口输入命令：

```
SELECT * FROM 学生 WHERE 学号 NOT IN ;
(SELECT 学号 FROM 成绩 WHERE 成绩<80)
```

主屏幕显示的内容为：

| 学号   | 姓名  | 年龄 |
|------|-----|----|
| 1001 | 张明  | 18 |
| 1005 | 张晓霞 | 22 |
| 1007 | 杜振宇 | 21 |
| 1008 | 安宏  | 16 |



【例 4-10】查询有不及格成绩的学生信息。

在命令窗口输入命令：

```
SELECT * FROM 学生 WHERE 学号=ANY ;
(SELECT 学号 FROM 成绩 WHERE 成绩<60)
```

主屏幕显示的内容为：

| 学号   | 姓名 | 年龄 |
|------|----|----|
| 1004 | 刘庆 | 18 |

#### 4. 计算查询

SQL 语言是完备的，也就是说，只要数据是按关系方式存入数据库的，就能构造合适的 SQL 语句把它检索出来。SQL 不仅具有一般的检索能力，而且还可以进行计算方式的检索。例如，关于平均数、最高数和最低数的检索。

用于检索计算的函数主要有：COUNT(计数)、SUM(求和)、AVG(求平均值)、MAX(最大值)和 MIN(最小值)。

【例 4-11】找出学生选修的课程的门数。

```
SELECT COUNT (DISTINCT 课程号) FROM 成绩
```

【例 4-12】求出学生的最高成绩、最低成绩和平均成绩。

在命令窗口输入命令：

```
SELECT MAX (成绩) AS 最高成绩 , MIN (成绩) AS 最低成绩 , AVG (成绩) AS 平均成绩
FROM 成绩
```

主屏幕显示的内容为：

| 最高成绩 | 最低成绩 | 平均成绩 |
|------|------|------|
| 99   | 56   | 79   |

#### 5. 排序

使用 SQL 可以将查询的结果排序，排序的短语是 ORDER BY，具体格式如下。

ORDER BY 排序选项 1 [ASC | DESC][,排序选项 2 [ASC | DESC]...]

从中可以看出，查询的结果可以升序（默认）或降序排列，并且允许按一列或多列排序。

【例 4-13】按学生成绩的降序查询学生的成绩信息。

```
SELECT * FROM 成绩 ORDER BY 成绩 DESC
```

#### 6. 分组与计算查询

利用 GROUP BY 子句进行分组计算查询，还可以用 HAVING 短语进一步限定分组条件。

【例 4-14】计算查询每个学生的平均成绩。

在命令窗口输入命令：

```
SELECT 学号 , AVG (成绩) FROM 成绩 GROUP BY 学号
```

主屏幕显示的内容为：

| 学号   | AVG_成绩 |
|------|--------|
| 1001 | 95     |
| 1002 | 81     |
| 1003 | 81     |
| 1004 | 62     |
| 1005 | 84     |
| 1006 | 63     |
| 1007 | 88     |
| 1009 | 67     |

【例 4-15】计算查询选修人数在 3 人以上的各科平均成绩。

在命令窗口输入命令：

```
SELECT 课程号, AVG(成绩) FROM 成绩 GROUP BY 课程号 HAVING COUNT(*) >= 3
```

主屏幕显示的内容为：

| 课程号  | AVG_成绩 |
|------|--------|
| KC02 | 77     |
| KC03 | 70     |
| KC05 | 86     |

7. 集合的并运算

SQL 支持集合的并 (UNION) 运算，即将两个 SELECT 语句的查询结果通过并运算合并成一个查询结果。为了进行并运算，要求两个查询的结果有相同的字段个数，并且对应字段的值要出自同一个值域。例如，下列语句的结果是“张明”和“刘庆”的信息。

```
SELECT * FROM 学生 WHERE 姓名="张明";
UNION;
SELECT * FROM 学生 WHERE 姓名="刘庆"
```

8. SQL 语言的几个特殊选项

(1) 显示部分结果。有时只需要满足条件的前几个记录，这时使用 TOP 数值表达式 PERCENT 短语非常有用。不使用 PERCENT 时，说明显示前几个记录；使用 PERCENT 时，表示显示前百分之几的记录。需要注意的是，TOP 短语与 ORDER BY 同时使用时才有效。

(2) 查询去向。

INTO 目的地 | TO FILE 文件名[ADDITIVE] | TO PRINT | TO SCREEN

用于把查询的结果送入不同目的地或外部设备，以适应不同用户的需要。默认的输出方式为浏览窗口。

INTO ARRAY 数组名：把查询结果填充到一个二维数组中。

INTO CURSOR 临时表文件名：把查询结果存储到一个临时表中，查询命令结束后，





临时表仍处于打开状态且是当前表，但它是只读的，一旦关闭临时表，临时表便立即被删除。

INTO DBF/TABLE 表文件名：把查询的结果存储到指定的表文件中，在磁盘上生成一个永久表文件。查询命令结束后，该表仍处于打开状态且是当前表。

TO FILE 文件名[ADDITIVE]：把查询结果存储到一个文本文件中，如果文件名为旧有的文件名且使用了 ADDITIVE 关键字，则把结果追加到该文件的尾部。

TO PRINT：把查询结果直接送往打印机。

TO SCREEN：把查询结果直接送往 Visual FoxPro 的主窗口。

#### 4.1.5 SQL 操作功能

##### 1. 插入

语句格式：

INSERT INTO 表名[(列名 1[,列名 2]...)] VALUES (表达式 1[,表达式 2]...);

该语句向表中插入一条新记录。

例如，向学生表中添加一条记录。

```
INSERT INTO 学生(学号,姓名,年龄) VALUES("1010","孙芳",18)
```

注意，当 VALUES 后面的表达式的数据类型、顺序和数目与表结构中字段的数据类型、顺序和数目完全一致时，表名后面的字段名可以省略。那么上面的插入语句可省略为：

```
INSERT INTO 学生 VALUES("1010","孙芳",18)
```

##### 2. 更新

语句格式：

UPDATE 表名

SET 列名 1=表达式 1[,列名 2=表达式 2]...

[WHERE 条件表达式];

该语句从指定表中修改满足条件表达式的元组，按 SET 子句中的表达式修改这些元组相应列上的值。

例如，将学号为 1001 的记录的姓名更新为“孙静”。

```
UPDATE 学生 SET 姓名="孙静" WHERE 学号="1001"
```

##### 3. 删除

语句格式：

DELETE FROM 表名 [WHERE 条件表达式];

该语句从指定的表中删除满足条件表达式的元组。如果没有条件子句，则删除所有的元组。

例如，从学生表中删除学号为 1006 的记录。

```
DELETE FROM 学生 WHERE 学号="1006"
```

注意：在 Visual FoxPro 中，SQL DELETE 命令同样是逻辑删除记录，若要物理删除，应继续使用 PACK 命令。

#### 4.1.6 SQL 定义功能

SQL 语言的数据定义（DDL）功能包括 3 个部分：定义基表、定义视图和定义索引。其中定义基表又包括建立基表、修改基表和删除基表；定义视图中包括建立视图、修改视图和删除视图；定义索引包括建立索引和删除索引。本节主要介绍 Visual FoxPro 支持的表定义和视图定义功能。

##### 1. 表的定义

建立基表的语句格式：

```
CREATE TABLE 表名(列名 1 数据类型 1[NOT NULL],[列名 2 数据类型 2[NOT NULL]]...)
```

[IN 数据库空间名];

一个表可以由一个列或多个列构成，列的定义要说明列名、数据类型，并指出列的值是否允许为空值（NULL）。如果某列作为该表的关键字，应该定义该列为非空（NOT NULL）。

例如，定义以上操作的表“成绩”。

```
CREATE TABLE 成绩(学号 C(4),课程号 C(4),成绩 I CHECK(成绩>=0 AND 成绩<=100)ERROR "成绩应在 0 到 100 之间")
```

在这里用 CHECK 子句为成绩字段定义了有效性规则，用 ERROR 子句为该有效性规则定义了错误提示信息。

##### 2. 表的删除

删除基表的语句格式：

```
DROP TABLE 表名;
```

例如，删除基表“个人信息表”。

```
DROP TABLE 个人信息表
```

该语句把一个基表的定义，连同表中的所有数据记录、索引及由此表导出的所有视图全部删除，并释放相应的存储空间。

##### 3. 表结构的修改

```
ALTER TABLE 表名 ADD | ALTER 列名数据类型(宽度);
```

例如，给学生表添加一个“性别”字段，同时设置字段的有效性规则。



```
ALTER TABLE 学生 ADD 性别 C(2) CHECK 性别="男" OR 性别="女"
```

例如，将学生表的“学号”字段的宽度由原来的4改为6。

```
ALTER TABLE 学生 ALTER 学号 C(6)
```

#### 4. 视图的定义

在 Visual FoxPro 中视图是一个定制的虚拟表，可以是本地的、远程的或带参数的。视图要引用一个或多个表，或者引用其他视图，视图是可以更新的，它可以引用远程表。

建立基表的语句格式：

```
CREATE VIEW 视图名(字段名 [, 字段名...]) AS 查询语句
```

```
[WITH CHECK OPTION];
```

例如，以查询为基础建立“个人信息视图”。

```
CREATE VIEW 个人信息视图 AS SELECT * FROM STUDENT;
```

删除视图的语句格式：

```
DROP VIEW 视图名;
```

例如，删除“个人信息视图”。

```
DROP VIEW 个人信息视图;
```

## 任务 4.2 实例训练

### 4.2.1 启动查询设计器

利用查询设计器可以细致地创建查询，在 Visual FoxPro 9.0 系统中，可以用两种方法启动查询设计器。

#### 1. 菜单方式

操作步骤如下。

(1) 打开数据库文件“学籍管理数据库.dbc”，进入“数据库设计器”窗口，如图 4-1 所示。

(2) 在 Visual FoxPro 9.0 系统主菜单下，选择“文件”|“新建”命令，打开“新建”对话框，如图 4-2 所示。

(3) 在“新建”对话框中选中“查询”单选按钮，单击“新建”按钮，打开“添加表或视图”对话框，如图 4-3 所示。

(4) 在“添加表或视图”对话框中，选择要使用的表。如果是单表查询，则添加相应表后单击“关闭”按钮；如果是多表查询，则添加多个表后关闭。如选择 students 表，依次单击“添加”、“关闭”按钮后，进入“查询设计器”窗口，如图 4-4 所示。

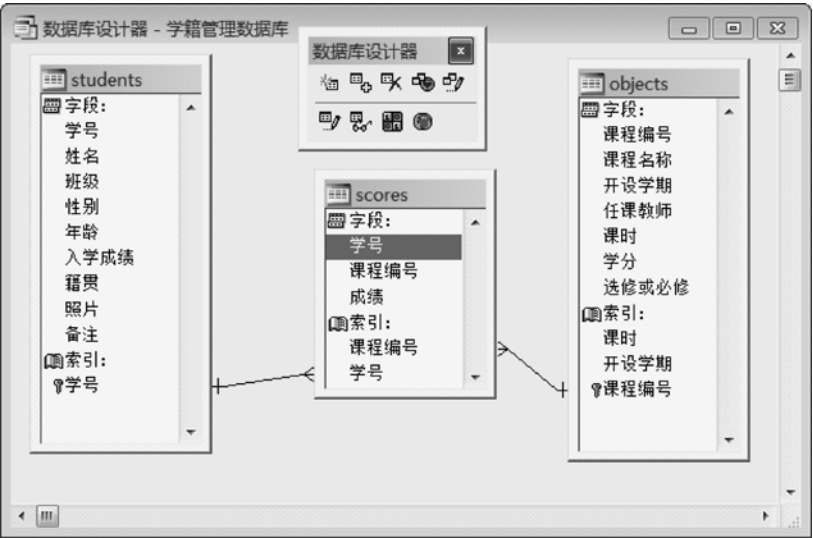


图 4-1 “数据库设计器”窗口



图 4-2 “新建”对话框

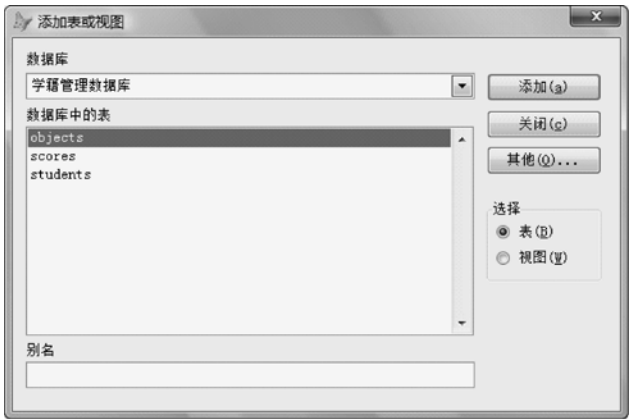


图 4-3 “添加表或视图”对话框



图 4-4 “查询设计器”窗口



## 2. 命令方式

在“命令”窗口中输入 CREATE QUERY 命令，也可进入“查询设计器”窗口，如图 4-5 所示。

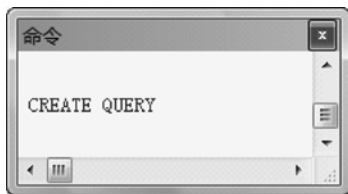


图 4-5 “命令”窗口

### 4.2.2 建立单表查询和多表查询

进入“查询设计器”窗口后的工作就是建立查询。

在查询设计器下方有 6 个选项卡，如图 4-6 所示，可以对查询的结果进行定义。下面以实例来说明建立查询的过程。



图 4-6 查询设计器下方的 6 个选项卡

#### 1. 单表查询

操作步骤如下。

(1) 在“Fields (字段)”选项卡的“可用字段”列表框中，选择希望在查询结果中出现的字段，单击“添加”按钮。如果希望所有字段都出现在查询结果中，可单击“添加全部”按钮。选择的字段将出现在“已选择字段”列表框中。在“已选择字段”列表框中选择字段，再单击“移除”按钮，可取消对字段的选择；若单击“全部移除”按钮，则取消对全部字段的选取。

(2) 要查询的字段选定之后，单击“查询设计器”窗口的关闭按钮，弹出如图 4-7 所示的系统提示对话框。



图 4-7 系统提示对话框

(3) 单击“是”按钮，弹出“另存为”对话框，如图 4-8 所示。



图 4-8 “另存为”对话框

(4) 输入创建查询的名称“学生信息查询.qpr”，单击“保存”按钮，一个查询文件即创建完成。

## 2. 多表查询

利用查询设计器不仅可以创建单表查询，还可以创建多表查询，即查询结果中的字段来自多个表。

操作步骤如下。

(1) 打开数据库文件“学籍管理数据库.dbc”，进入“数据库设计器”窗口。

(2) 在 Visual FoxPro 9.0 系统主菜单下，选择“文件”|“新建”命令，弹出“新建”对话框。

(3) 在“新建”对话框中选中“查询”单选按钮，单击“新建”按钮，弹出“添加表或视图”对话框。

(4) 选择学生信息表 students 与成绩表 scores，单击“关闭”按钮，则弹出“Join Condition (连接条件)”对话框，如图 4-9 所示。

(5) 依次单击“OK (确定)”按钮、“关闭”按钮，进入“查询设计器”窗口，如图 4-10 所示。

(6) 在“查询设计器”窗口的“可用字段”列表框中，依次将相关的字段添加到“已选择字段”列表框中，如图 4-11 所示。

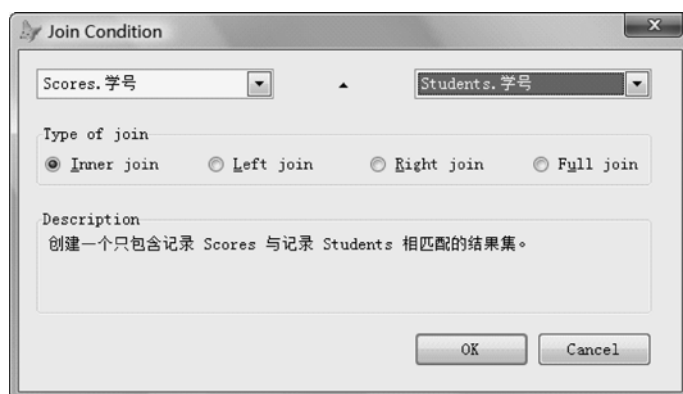


图 4-9 “Join Condition (连接条件)”对话框



图 4-10 “查询设计器”窗口

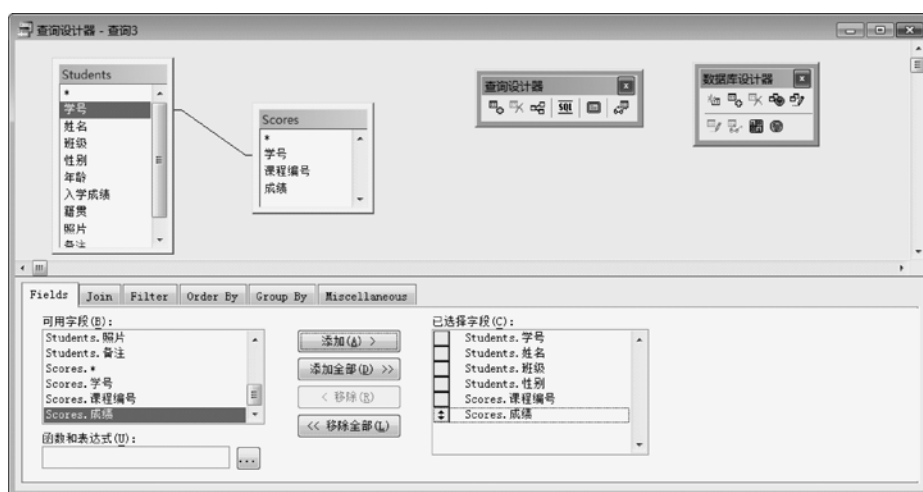


图 4-11 添加字段的操作

(7) 当两个表中可以出现在查询中的字段被选定以后，单击“查询设计器”窗口的关闭按钮，弹出系统提示对话框，如图 4-12 所示。



图 4-12 系统提示对话框

(8) 单击“是”按钮，弹出“另存为”对话框，如 4-13 所示。



图 4-13 “另存为”对话框

(9) 输入创建查询的名字“学生信息与成绩查询”，单击“保存”按钮，又一个查询文件即创建完成。

#### 4.2.3 运行查询

运行查询就是确定查询结果的输出格式，比使用数据表、数据库表和数据库更方便、快捷和安全。

下面就利用已有查询文件定制查询结果的输出格式，举例说明操作方法，操作步骤如下。

(1) 选择“文件”|“打开”命令，弹出“打开”对话框，如图 4-14 所示。

(2) 选择相应的查询文件，单击“确定”按钮。

(3) 进入“查询设计器”窗口后，在 Visual FoxPro 9.0 系统主菜单下，选择“查询”|“查询去向”命令，弹出“Query Destination (查询去向)”对话框，如图 4-15 所示。

(4) 系统在“Query Destination (查询去向)”对话框中，提供了以下 4 种输出格式，决定查询结果的输出方式。

“Browse (浏览)”格式：将查询结果送入浏览窗口。

“Cursor (临时表)”格式：将查询结果存入一个临时的数据表中，用户可以随意处理该临时表。





图 4-14 “打开”对话框

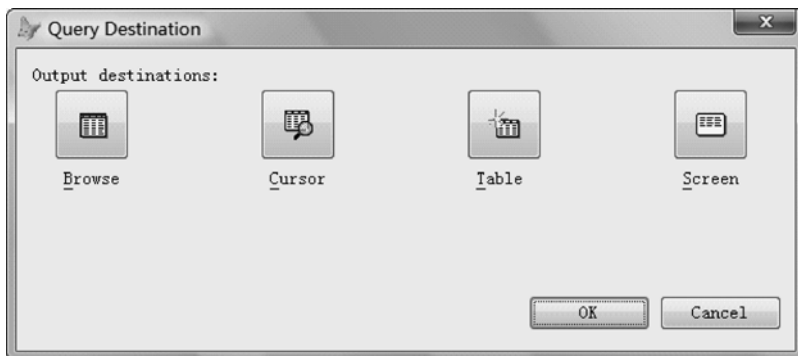


图 4-15 “Query Destination (查询去向)”对话框

“Table (表)”格式：将查询结果存入一个数据表中，用户可以随意处理该数据表，当关闭该数据表后，查询结果仍存在磁盘上。

“Screen (屏幕)”格式：将查询结果输出到屏幕上。

#### 4.2.4 利用视图设计器创建单表本地视图

利用视图设计器来创建本地视图，现举例说明操作方法。

例如，利用视图设计器，依据“学籍管理数据库.dbc”，创建一个单表本地视图。

操作步骤如下。

(1) 打开数据库文件“学籍管理数据库.dbc”，进入“数据库设计器”窗口，如图 4-16 所示。

(2) 在 Visual FoxPro 9.0 系统主菜单下，选择“文件”|“新建”命令，弹出“新建”对话框，如图 4-17 所示。

(3) 在“新建”对话框中选中“视图”单选按钮，单击“新建”按钮，弹出“添加表或视图”对话框，如图 4-18 所示。

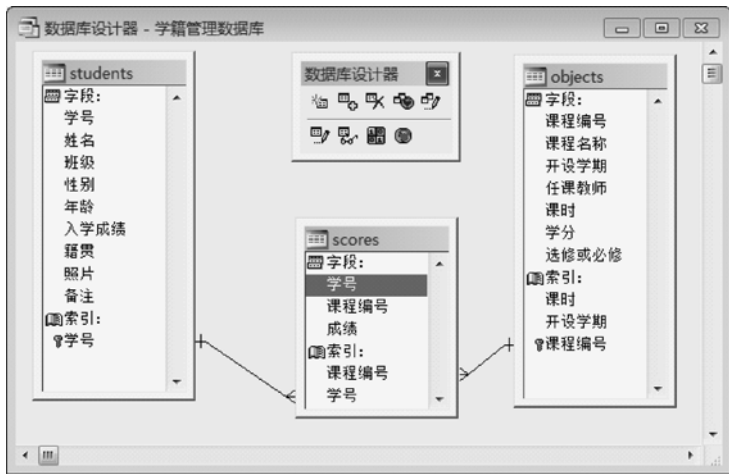


图 4-16 “数据库设计器”窗口



图 4-17 “新建”对话框

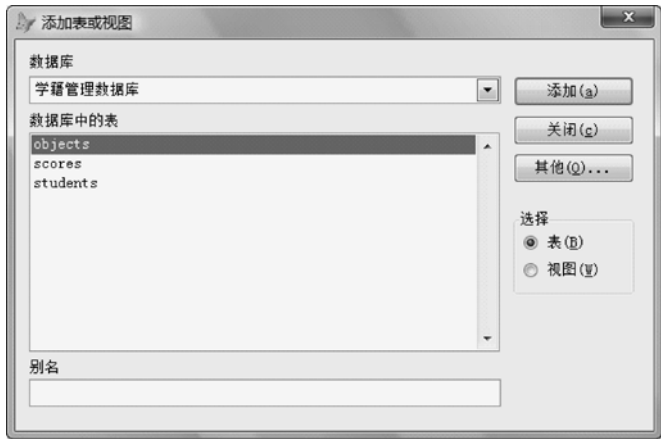


图 4-18 “添加表或视图”对话框

(4) 在“添加表或视图”对话框中，把建立视图所依据的相关表添加到视图设计器中，再单击关闭按钮，进入视图设计器，如图 4-19 所示。



图 4-19 视图设计器



(5) 在视图设计器的“可用字段”列表框中，逐个单击可用的相关字段，将其添加到“选定字段”列表框中。

(6) 当表中可以出现在视图中的字段被选定后，单击“视图设计器”窗口的关闭按钮，弹出系统提示对话框。

(7) 单击“是”按钮，弹出“保存”对话框，如图 4-20 所示。

(8) 输入创建视图的名称“学生信息视图”，单击“确定”按钮，一个视图文件即建立完成，同时被存放在打开的数据库中，如图 4-21 所示。



图 4-20 “保存”对话框

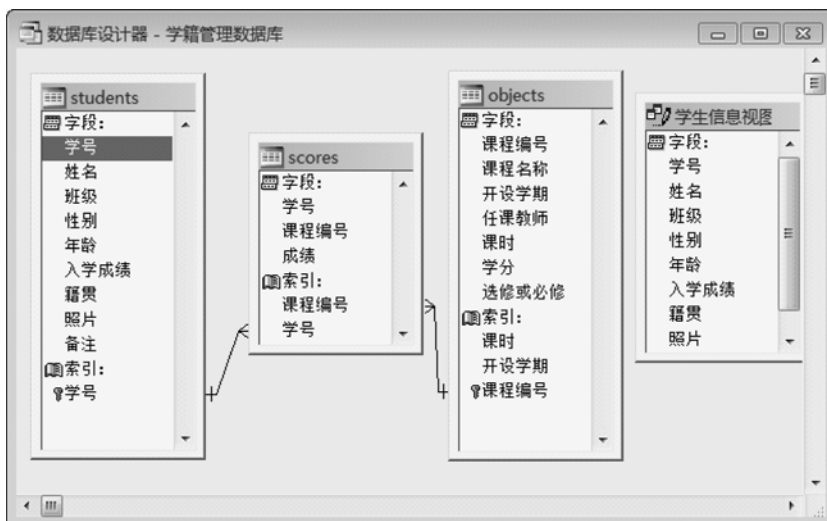


图 4-21 新建视图文件的存放

(9) 选择“数据库”|“浏览”命令，进入视图浏览窗口，如图 4-22 所示。

| Students |     |         |    |    |     |      |     |   |
|----------|-----|---------|----|----|-----|------|-----|---|
| 学号       | 姓名  | 班级      | 性别 | 年龄 | 入学成 | 籍贯   | 照片  |   |
| 0104001  | 陈达文 | 09高职计信管 | 男  | 23 | 345 | 新疆奎屯 | gen | f |
| 0104002  | 陈桂芬 | 09高职计信管 | 女  | 18 | 240 | 新疆伊犁 | gen | f |
| 0104003  | 江培强 | 09高职计信管 | 男  | 22 | 420 | 广东高州 | gen | f |
| 0104004  | 林江文 | 09高职计信管 | 男  | 22 | 380 | 广东湛江 | gen | f |
| 0104005  | 林世鑫 | 09高职计信管 | 男  | 26 | 290 | 广东徐闻 | gen | f |
| 0104006  | 陈玉强 | 09高职计信管 | 男  | 22 | 315 | 广东湛江 | gen | f |
| 0104007  | 谭旭全 | 09高职多媒体 | 男  | 22 | 300 | 广东廉江 | gen | f |
| 0041001  | 洪明信 | 09高职多媒体 | 男  | 23 | 320 | 广东信宜 | gen | f |
| 0041002  | 李小华 | 09高职多媒体 | 女  | 23 | 310 | 广东揭阳 | gen | f |

图 4-22 视图浏览窗口

从图中可以看出，学生信息视图中的数据是 students 表中的数据。

#### 4.2.5 利用视图设计器创建多表本地视图

利用视图设计器，依据“学籍管理数据库.dbc”，创建一个多表本地视图。

操作步骤如下。

- (1) 打开数据库文件“学籍管理数据库.dbc”，进入“数据库设计器”窗口。
- (2) 选择“数据库”|“新建本地视图”命令，弹出“新建本地视图”对话框。
- (3) 单击“新建视图”按钮，进入“视图设计器”窗口，同时弹出“添加表或视图”对话框。
- (4) 在“添加表或视图”对话框中，把建立视图所依据的表 students 和 scores 添加到视图设计器中，再打开“Join Condition (连接条件)”对话框，如图 4-23 所示。

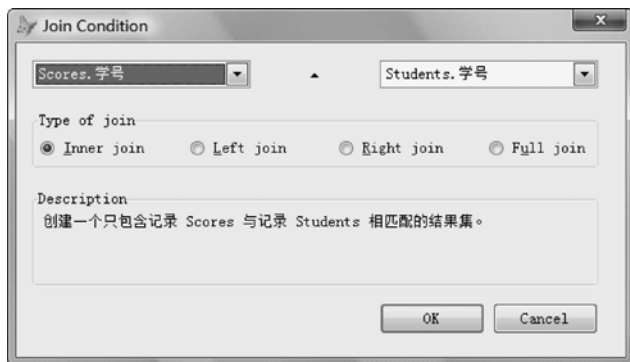


图 4-23 “Join Condition (连接条件)”对话框

- (5) 选择表间的连接类型，单击“OK (确定)”按钮，返回“视图设计器”窗口。
- (6) 在视图设计器的“可用字段”列表框中，逐个单击可用的相关字段，把相关字段添加到“已选择字段”列表框中，如图 4-24 所示。



图 4-24 添加字段的操作

- (7) 当表中可以出现在视图中的字段被选定后，单击“视图设计器”窗口的关闭按钮，弹出系统提示对话框。



(8) 单击“是”按钮，弹出“保存”对话框，如图 4-25 所示。



图 4-25 “保存”对话框

(9) 输入创建视图的名称“学生信息与成绩视图”，单击“确定”按钮，一个视图文件即建立完成，同时被存放在打开的数据库中，如图 4-26 所示。



图 4-26 新建视图文件的存放

(10) 选择“数据库”|“浏览”命令，进入视图浏览窗口，如图 4-27 所示。

| 学号      | 姓名  | 班级      | 性别 | 年龄 | 课程编号   | 成绩 |
|---------|-----|---------|----|----|--------|----|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | jsj005 | 82 |
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | jsj002 | 76 |
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | jsj001 | 80 |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | jsj006 | 86 |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | jsj003 | 90 |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | jsj001 | 88 |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | jsj007 | 93 |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | isi002 | 98 |

图 4-27 视图浏览窗口

从图 4-27 中可以看出, 学生信息与成绩视图中的数据是 students 表与 scores 表中的部分字段的重新组合。

#### 4.2.6 利用视图向导创建单表本地视图

创建视图的另一种方法是利用视图向导, 现将操作方法举例说明。

例如, 利用视图向导, 根据“学籍管理数据库.dbc”创建一个单表本地视图“课程信息视图”。

操作步骤如下。

(1) 打开数据库文件“学籍管理数据库.dbc”, 进入“数据库设计器”窗口。

(2) 在 Visual FoxPro 9.0 系统主菜单下, 选择“文件”|“新建”命令, 弹出“新建”对话框, 选中“视图”单选按钮。

(3) 单击“向导”按钮, 进入“Local View Wizard (本地视图向导)”步骤 1 对话框, 如图 4-28 所示。

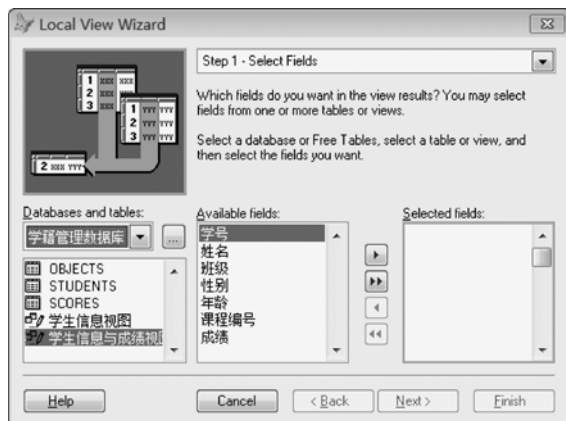


图 4-28 “Local View Wizard (本地视图向导)”步骤 1 对话框

(4) 在“Local View Wizard (本地视图向导)”对话框中, 选择课程信息表 objects 为数据来源, 然后逐个选择要出现在视图中的字段, 再单击“Next(下一步)”按钮, 进入“Local View Wizard (本地视图向导)”步骤 3 对话框, 如图 4-29 所示。

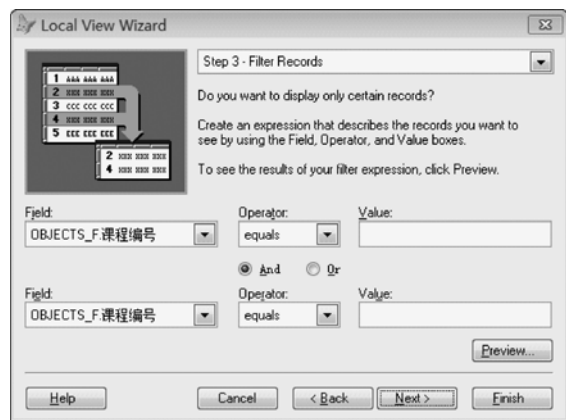


图 4-29 “Local View Wizard (本地视图向导)”步骤 3 对话框



(5) 单击“Next (下一步)”按钮, 进入步骤 4 对话框, 如图 4-30 所示。



图 4-30 “Local View Wizard (本地视图向导)” 步骤 4 对话框

(6) 在“Local View Wizard (本地视图向导)”步骤 4 对话框中单击“Next (下一步)”按钮, 进入步骤 4a 对话框, 如图 4-31 所示。

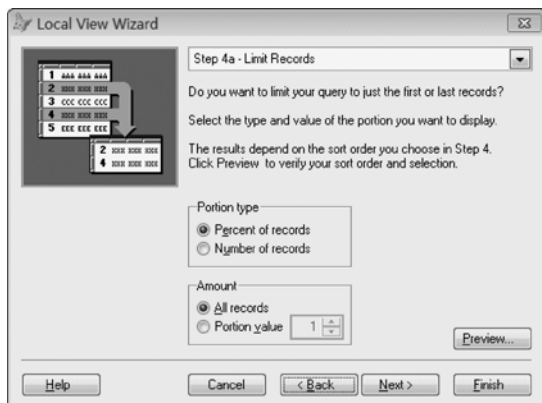


图 4-31 “Local View Wizard (本地视图向导)” 步骤 4a 对话框

(7) 在“Local View Wizard (本地视图向导)”步骤 4a 对话框中单击“Next (下一步)”按钮, 进入步骤 5 对话框, 如图 4-32 所示。

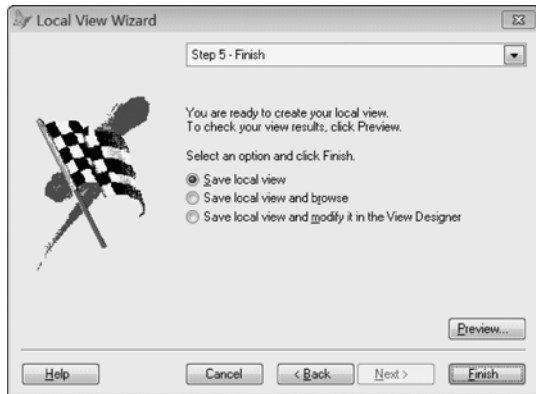


图 4-32 “Local View Wizard (本地视图向导)” 步骤 5 对话框

(8) 在“Local View Wizard(本地视图向导)”步骤5对话框中,当选中“Save local view(保存本地视图)”单选按钮后,再单击“Finish(完成)”按钮,弹出“View Name(视图名称)”窗口,如图4-33所示。



图 4-33 “View Name(视图名称)”窗口

(9) 输入创建视图的名称,单击“OK(确定)”按钮,返回 Visual FoxPro 系统主菜单下。此时,利用视图向导创建单表本地视图操作已结束,如图4-34所示。



图 4-34 操作完成后的主菜单

(10) 选择“数据库”|“浏览”命令,进入浏览窗口,如图4-35所示。

| 课程编号   | 课程名称     | 开设学期 | 任课教师 | 课时 | 学分 | 选修 |
|--------|----------|------|------|----|----|----|
| jsj001 | 操作系统     | 第一学期 | 赵玉   | 50 | 5  | 必修 |
| jsj002 | VB语言     | 第一学期 | 钱红   | 45 | 4  | 必修 |
| jsj003 | 计算机基础    | 第一学期 | 孙南   | 45 | 2  | 必修 |
| jsj004 | 计算机网络    | 第二学期 | 郑东   | 40 | 3  | 选修 |
| jsj005 | 面向对象技术   | 第三学期 | 李兰   | 45 | 4  | 必修 |
| jsj006 | Delphi应用 | 第三学期 | 周五   | 50 | 3  | 必修 |

图 4-35 浏览窗口

利用视图向导创建一个多表本地视图与创建单表本地视图类似,这里不再做过多说明。





#### 4.2.7 利用本地视图更新数据表中的数据

虽然视图是个“虚表”，但是也可以利用视图更新数据。因为视图可以限定数据表上数据的使用范围，因此，也就限定了可更新数据，数据表中其他数据就不会被破坏，因此可以提高数据维护的安全性。

下面举例说明利用本地视图更新数据表中的数据。

例如，利用本地视图“学生信息与成绩视图”更新 students 表和 scores 表中的相关数据。操作步骤如下。

(1) 打开数据库文件“学籍管理数据库.dbc”，进入“数据库设计器”窗口，如图 4-36 所示，激活“学生信息与成绩”视图。



图 4-36 “数据库设计器”窗口

(2) 选择“数据库”|“修改”命令，进入“视图设计器”窗口，如图 4-37 所示。

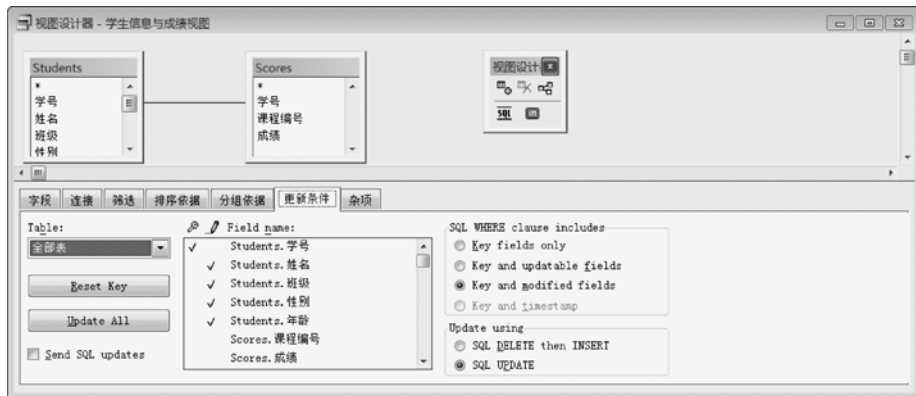


图 4-37 “视图设计器”窗口

(3) 在“视图设计器”中,选择“更新条件”选项卡。



(4) 在这个选项卡中,“Field name(字段名)”列表框中的图标“”表示关键字段,图标“”表示可更新字段。单击字段名前面的位置,可以设置或取消。如果选中“Send SQL updates(发送 SQL 更新)”复选框,可把视图的更新结果返回到表中。当要更新的字段名确定后,单击关闭按钮,弹出系统提示对话框,如图 4-38 所示。



图 4-38 系统提示对话框

(5) 保存修改并返回到 Visual FoxPro 9.0 主窗口,就可以利用学生信息与成绩视图更新数据表中要更新的字段的数据了。

## 任务 4.3 项目分析

在这个项目中,我们建立了两个查询和三个视图,依次为学生信息查询、学生信息与成绩查询、学生信息视图、学生信息与成绩视图和课程信息视图。其中学生信息查询是一个单表查询,学生信息与成绩查询是一个多表查询,学生信息视图和课程信息视图是单表视图,而学生信息与成绩视图是一个多表视图。下面我们来依次分析这些查询和视图。

### 4.3.1 学生信息查询

选择“文件”|“打开”命令,弹出打开对话框,在“学生学籍管理系统”中找到“学生信息查询.qpr”,单击“确定”按钮,进入学生信息查询的“查询设计器”窗口,如图 4-39 所示。



图 4-39 学生信息查询的“查询设计器”窗口



选择查询设计器中的“SQL”快捷命令，可以看到该查询的命令窗口，如图 4-40 所示。



图 4-40 学生信息查询的命令窗口

### 4.3.2 学生信息与成绩查询

选择“文件”|“打开”命令，弹出打开对话框，在“学生学籍管理系统”中找到“学生信息与成绩查询.qpr”，单击“确定”按钮，进入学生信息与成绩查询的“查询设计器”窗口，如图 4-41 所示。

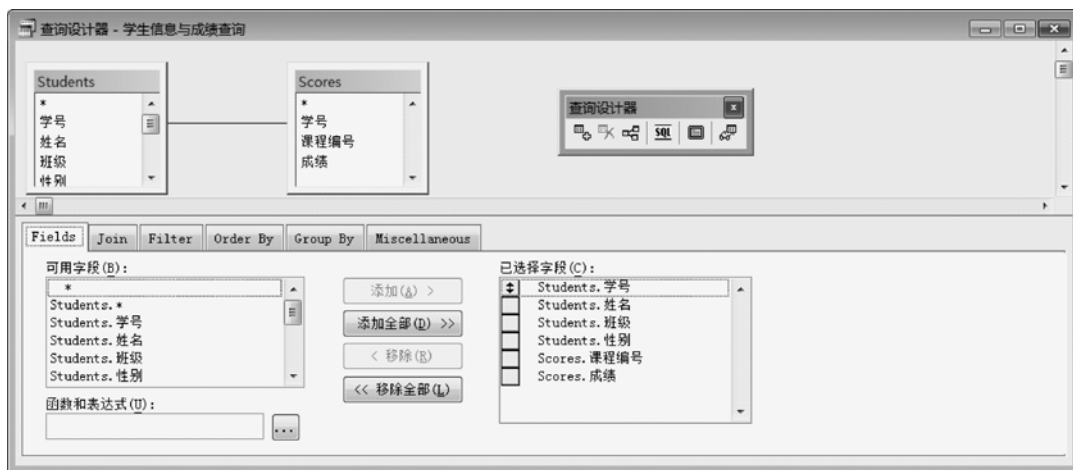


图 4-41 学生信息与成绩查询的“查询设计器”窗口

选择查询设计器中的“SQL”快捷命令，可以看到该查询的命令窗口，如图 4-42 所示。

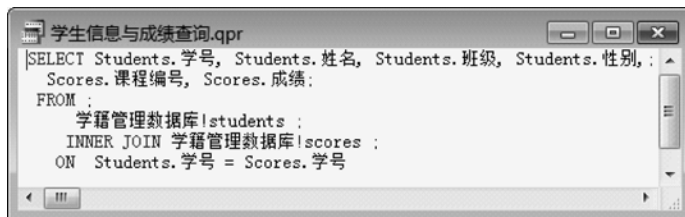


图 4-42 学生信息与成绩查询的命令窗口

### 4.3.3 学生信息视图

打开“学籍管理数据库”的数据库设计窗口，可以看到之前建立的视图文件，学生信息视图，如图 4-43 所示。

在学生信息视图空白处单击右键，选择“修改”命令，进入视图设计器窗口。选择查询设计器中的“SQL”快捷命令，可以看到该查询的命令窗口，如图 4-44 所示。



图 4-43 学生信息视图

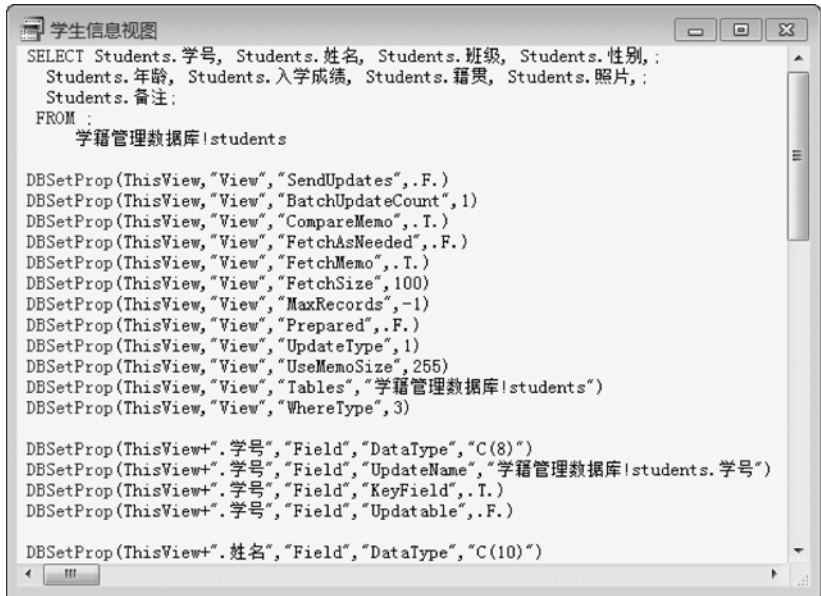


图 4-44 学生信息视图的命令窗口

#### 4.3.4 学生信息与成绩视图

打开“学籍管理数据库”的数据库设计窗口，可以看到之前建立的视图文件，学生信息与成绩视图，如图 4-45 所示。

在学生信息与成绩视图空白处单击右键，选择“修改”命令，进入视图设计器窗口。选择查询设计器中的“SQL”快捷命令，可以看到该查询的命令窗口，如图 4-46 所示。

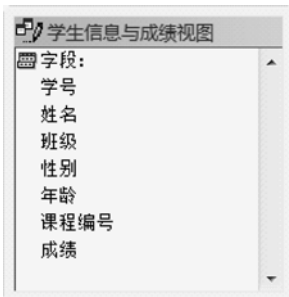


图 4-45 学生信息与成绩视图

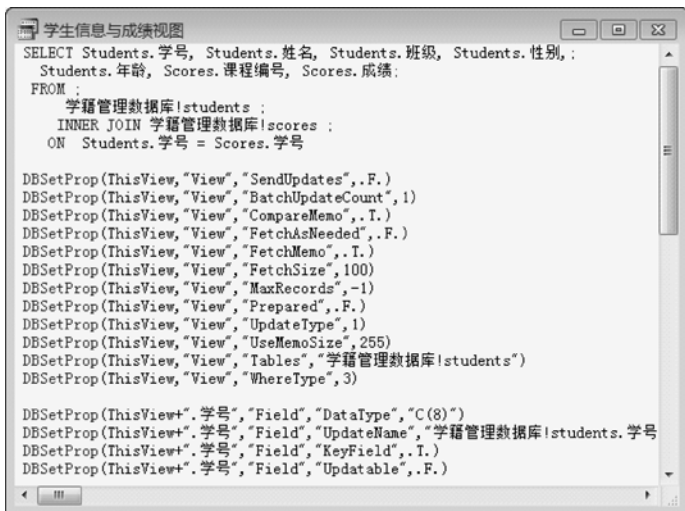


图 4-46 学生信息与成绩视图的命令窗口



### 4.3.5 课程信息视图

打开“学籍管理数据库”的数据库设计窗口，可以看到之前建立的视图文件，课程信息视图如图 4-47 所示。

在课程信息视图空白处单击右键，选择“修改”命令，进入视图设计器窗口。选择查询设计器中的“SQL”快捷命令，可以看到该查询的命令窗口，如图 4-48 所示。



图 4-47 课程信息视图

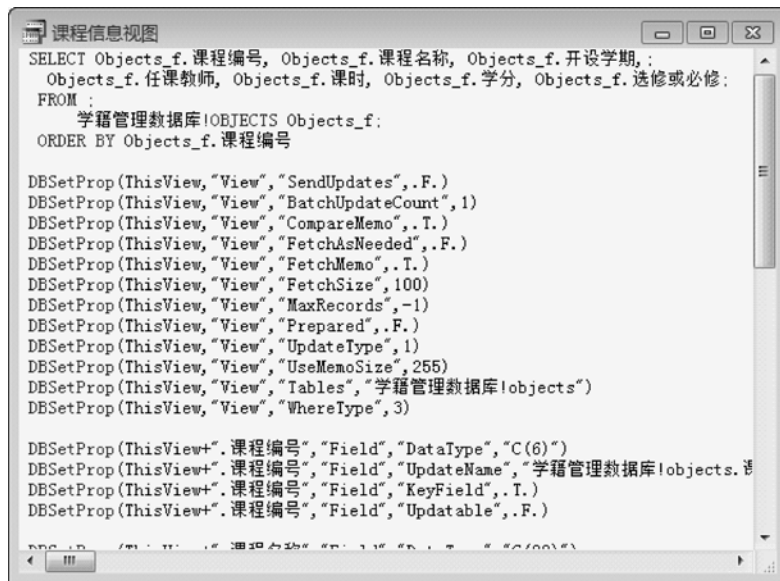


图 4-48 课程信息视图的命令窗口

## 任务 4.4 模仿设计与自主设计

### 4.4.1 建立查询文件

(1) 利用图书信息表，使用菜单方式建立查询文件 q1，显示数量为“0”的图书的编号、名称、出版日期、价格和数量。

(2) 利用借阅者信息表，使用查询设计器建立查询文件 q2，显示男性借阅者的借书证号、姓名、身份证和联系电话。

(3) 利用图书信息表、借阅者信息表和租借信息表，使用查询设计器建立查询文件 q3，显示图书编号、图书名称、借书证号、姓名和联系电话。

### 4.4.2 建立视图文件

(1) 利用图书信息表，使用向导建立“图书信息视图”，显示数量为“0”的图书的编号、名称、出版日期、价格和数量。

(2) 利用借阅者信息表，使用视图设计器建立“借阅者视图”，显示男性借阅者的借书

证号、姓名、身份证和联系电话。

(3) 利用图书信息表、借阅者信息表和租借信息表,使用视图设计器建立“租借信息视图”,显示图书编号、图书名称、借书证号、姓名和联系电话。

## 项目小结 4

查询与视图的操作在 Visual FoxPro 中经常用到。项目 4 主要介绍了查询与视图的基础知识与使用方法。用户可以方便地使用查询设计器与视图设计器来对数据进行检索,并且可以从一个或多个表中检索数据,同时可设置各种检索条件,并对查询结果进行排序与分组,输出到多种对象中。

视图与查询有许多相似之处,但也有本质的区别:第一,视图可以更新数据,查询不能更新数据;第二,查询保存在独立的文件中,而视图不是独立的文件,它只能存储在数据库中。

项目 4 还介绍了 SQL 检索。SQL 已成为 Visual FoxPro 的基本内容,要用好 Visual FoxPro 不掌握 SQL 是不行的。可以说,掌握 SQL 是学好、用好 Visual FoxPro 的基础。

## 项目习题 4

### 1. 选择题

- (1) 下列有关查询和视图的说法,正确的是( )。
- A. 查询设计器中有“更新条件”选项卡,查询有多种查询去向  
B. 查询设计器中没有“更新条件”选项卡,查询没有多种查询去向  
C. 查询设计器中没有“更新条件”选项卡,视图没有多种查询去向  
D. 查询设计器中有“更新条件”选项卡,视图没有多种查询去向
- (2) 下列各项中,哪一项不是以 PRG 为扩展名的( )。
- A. 子程序      B. 过程文件      C. 独立自定义函数      D. 项目文件
- (3) 关于视图,下列说法错误的是( )。
- A. 对视图的使用与表一样,也可以进行插入、查询、删除和修改操作  
B. 视图与查询一样,都以文件的形式独立存在,都可以为数据库所管理  
C. 视图与表不一样,它是一种虚表,不存储数据  
D. 在 VFP 中,可建立本地视图和远程视图
- (4) 利用查询设计器创建的查询,其查询结果输出去向的默认类型是( )。
- A. 临时表      B. 浏览窗口      C. 表      D. 屏幕
- (5) 在 SQL 查询时,使用 WHERE 子句指出的是( )。
- A. 查询目标      B. 查询结果      C. 查询条件 Q      D. 查询视图
- (6) 下面关于查询描述正确的是( )。
- A. 可以使用 CREATE VIEW 打开查询设计器



- B. 使用查询设计器可以生成所有的 SQL 查询语句
- C. 使用查询设计器生成的 SQL 语句存盘后将存放在扩展名为 QPR 的文件中
- D. 使用 DO 语句执行查询时, 可以不带扩展名

(7) 视图设计器中包含的选项卡有 ( )。

- A. 连接、显示、排序
- B. 更新条件、排序条件、显示
- C. 显示、排序依据、分组依据
- D. 更新条件、筛选、字段

(8) 在 VFP 中, 建立多表查询时, 默认表间的连接类型是 ( )。

- A. 内部
- B. 左连接
- C. 右连接
- D. 完全连接

(9) 使用 SQL 语句进行分组检索时, 为了去掉不满足条件的分组, 应当 ( )。

- A. 使用 WHERE 子句
- B. 在 GROUP BY 后面使用 HAVING 子句
- C. 先使用 WHERE 子句, 再使用 HAVING 子句
- D. 先使用 HAVING 子句, 再使用 WHERE 子句

(10) 在 Visual FoxPro 中, 使用 SQL 命令将学生表 STUDENT 中的学生年龄 AGE 字段的值增加 1 岁, 应该使用的命令是 ( )。

- A. REPLACE AGE WITH AGE+1
- B. UPDATE STUDENT AGE WITH AGE+1
- C. UPDATE SET AGE WITH AGE+1
- D. UPDATE STUDENT SET AGE=AGE+1

(11) 如果学生表 STUDENT 是使用下面的 SQL 语句创建的:

```
CREATE TABLE STUDENT(SNO C(4) PRIMARY KEY NOT NULL;
SN C(8);
SEX C(2);
AGE N(2) CHECK(AGE>15 AND AGE<30))
```

下面的 SQL 语句中可以正确执行的是 ( )。

- A. INSERT INTO STUDENT (SNO,SEX,AGE) values ("S9","男",17)
- B. INSERT INTO STUDENT (SNO,SEX,AGE) values ("李安琦","男",20)
- C. INSERT INTO STUDENT (SEX,AGE) values ("男",20)
- D. INSERT INTO STUDENT (SNO,SN) values ("S9","安琦",16)

(12) 使用 SQL 语句从表 STUDENT 中查询所有姓“王”的同学的信息, 正确的命令是 ( )。

- A. SELECT\*FROM STUDENT WHERE LEFT (姓名,2)="王"
- B. SELECT\*FROM STUDENT WHERE RIGHT (姓名,2)="王"
- C. SELECT\*FROM STUDENT WHERE TRIM (姓名,2)="王"
- D. SELECT\*FROM STUDENT WHERE STR (姓名,2)="王"

(13) 有 SQL 语句:

SELECT\*FROM 教师 WHERE NOT(工资>3000 OR 工资<2000)

与以上语句等价的 SQL 语句是 ( )。

- A. SELECT \* FROM 教师 WHERE 工资 BETWEEN 2000 AND 3000

- B . SELECT \* FROM 教师 WHERE 工资>2000 AND 工资<3000
- C . SELECT \* FROM 教师 WHERE 工资>2000 OR 工资<3000
- D . SELECT \* FROM 教师 WHERE 工资<=2000 AND 工资>=3000

(14) 使用 SQL 语句将学生表 S 中年龄 ( AGE ) 大于 30 岁的记录删除, 正确的命令是 ( )。

- A . DELETE FOR AGE > 30
- B . DELETE FROM S WHERE AGE > 30
- C . DELETE FOR AGE > 30
- D . DELETE S WHERE AGE > 30

(15) “图书”表中有字符型字段“图书号”。要求用 SQL DELETE 命令将图书号以字母 A 开头的图书记录全部打上删除标记, 正确的命令是 ( )。

- A . DELETE FROM 图书 FOR 图书号 LIKE "A%"
- B . DELETE FROM 图书 WHILE 图书号 LIKE "A%"
- C . DELETE FROM 图书 WHERE 图书号="A\*"
- D . DELETE FROM 图书 WHERE 图书号 LIKE "A%"

## 2. 填空题

(1) 在 SQL 的 CREATE TABLE 语句中, 为属性说明取值范围 ( 约束 ) 的是\_\_\_\_\_短语。

(2) SQL 插入记录的命令是 INSERT, 删除记录的命令是\_\_\_\_\_, 修改记录的命令是\_\_\_\_\_。

(3) 在 SQL 的嵌套查询中, 量词 ANY 和\_\_\_\_\_是同义词。在 SQL 查询时, 使用\_\_\_\_\_子句限定的是查询条件。

(4) 为“学生”表增加一个“平均成绩”字段的正确命令是: ALTER TABLE 学生 ADD \_\_\_\_\_平均成绩 N(5,2)。

(5) “职工”表有工资字段, 计算工资合计的 SQL 语句是: SELECT \_\_\_\_\_FROM 职工。

(6) 将“金牌榜.DBF”中新增加的字段“奖牌总数”设置为金牌数、银牌数、铜牌数 3 项之和, 应该使用 SQL 语句: \_\_\_\_\_金牌榜\_\_\_\_\_奖牌总数=金牌数+银牌数+铜牌数。





# 项目 5

## 学生学籍管理系统 程序设计

### 项目要点◎

---

- 程序文件的建立与执行。
- 程序的基本结构。
- 过程及过程文件。

### 学习目标◎

---

- 掌握创建和编辑程序文件的基本方法。
- 掌握结构化程序设计的基础知识。
- 掌握选择结构、循环结构的基本功能。
- 掌握程序设计常用命令的功能。
- 掌握使用过程的基本方法。

Visual FoxPro 系统不但拥有大量的交互式数据库管理工具，而且还拥有一整套功能完善的程序语言系统，以及过程式程序设计和面向对象的可视化程序编写工具。在 Visual FoxPro 系统环境下，交互式的数据库操作程序工作方式也可以完成。在实际应用中，程序方式是最重要的操作，也是最常用的方式，用这种方式可以调用 Visual FoxPro 系统功能，以完成更为复杂的任务。

本项目将介绍程序设计的基础知识，包括程序与程序文件、程序的基本结构、多模块程序及程序调试等。

## 任务 5.1 熟悉程序设计基本知识

### 5.1.1 程序的建立与编辑

在 Visual FoxPro 系统中,用命令对表进行操作可分为两种方式:交互方式(单命令方式)和程序方式(批命令方式),但是在对各类事务的管理过程中,如果使用单命令方式,就只能逐条输入、逐条执行各命令自身的功能,必然会出现重复输入相同命令的操作,导致执行效率低,解决问题的能力差。所以通常采用程序方式,按解题的要求和步骤,将各种不同功能的命令放在一起,成为一个“命令”程序。运行该程序,系统将自动执行程序中的每条命令,完成所要求的功能,这无疑将给我们的工作带来极大的方便。这种由各种命令集合而成的文件,即称为程序文件,扩展名为.prg。在 Visual FoxPro 系统环境下,通过菜单方式或命令方式运行该程序文件。

在编写程序时输入的每条命令称为命令语句,执行命令文件就是按照该文件即程序指定的顺序执行各命令语句。

#### 1. 建立程序文件

对于 Visual FoxPro,任何可以建立、编辑文本文件的工具,都可以创建和编辑 Visual FoxPro 程序文件,VFP 系统本身有极强大的编辑器,可用来编辑程序文件。

操作步骤如下。

(1) 选择“文件”“新建”命令,在弹出的“新建”对话框中选中“程序”单选按钮。

(2) 单击“新建”按钮,弹出 Visual FoxPro 文本编辑窗口。

(3) 在文本编辑窗口中输入程序内容。这里的编辑操作与普通文本文件的编辑操作没有什么不同,这里输入的是程序内容,是一条条指令。与命令窗口输入命令不同,这里输入的命令是不会被立即执行的。

(4) 保存程序文件。选择“文件”“保存”命令或按“Ctrl+W”组合键,在弹出的“另存为”对话框中指定程序的存放位置和文件名,单击“保存”按钮。

也可以用命令方式建立或修改程序文件。

命令格式为:MODIFY COMMAND<程序文件名>

这里的文件名前可以指定保存文件的路径。如果没有给定扩展名,系统会自动加上默认的扩展名.prg。

#### 2. 打开和修改程序文件

要打开、修改程序文件,可按下列方法操作。

(1) 选择“文件”“打开”命令,弹出“打开”对话框,在“文件类型”下拉列表框中选择“程序”类型。

(2) 在文件列表中选定要修改的文件,单击“确定”按钮。

(3) 编辑修改完成后,再进行保存。若要放弃本次修改,可从“文件”菜单中单击“还原”命令或按 Esc 键。



### 5.1.2 程序的调用与执行

调用程序文件,即运行程序文件,是程序文件建立的最终目的。调用程序文件的方法有以下两种。

#### 1. 菜单方式

- (1) 在 VFP 主菜单栏中,选择“程序” “运行”命令。
- (2) 在“运行”窗口中,输入被调用的程序文件名即可。

#### 2. 命令方式

在命令窗口中输入以下命令,则程序文件被调用。

命令格式:DO<文件名>

该命令可以在命令窗口中发出,也可以在某个程序中使用,这样就可以在一个程序中调用另一个程序。

当程序文件被执行时,文件中包含的命令将被依次执行,直到所有的命令被执行完毕,除非执行到以下命令。

- (1) CANCEL,终止程序运行,清除所有的私有变量,返回命令窗口。
- (2) DO,转去执行其他程序。
- (3) RETURN,结束当前程序的执行,返回到调用它的上级程序,若无上级程序则返回到命令窗口。
- (4) QUIT,退出 Visual FoxPro 系统,返回操作系统。

### 5.1.3 简单的输入/输出命令

VisualFoxPro 系统,提供了3种使用灵活、功能强大的人机对话式赋值命令:ACCEPT、WAIT、INPUT,以实现人工干预程序执行的目的。

#### 1. 键盘输入字符串命令

格式:ACCEPT [<字符表达式>] TO <内存变量>

功能:暂停程序的执行,接收用户从键盘上输入的字符串并赋值给内存变量,按 Enter 键结束输入。

说明如下。

- (1) 如果选用<字符表达式>,那么系统首先显示该字符表达式,作为提示信息。
- (2) 该命令只接受字符串,用户在输入字符串时,不需要加界定符,否则界定符也作为字符串的一部分。

#### 2. 键盘输入单字符命令

格式:WAIT [<提示信息>] [TO <内存变量>] [WINDOWS [AT <行>,<列>]]

功能:暂停程序的执行,接收用户从键盘输入的单字符。

说明如下。

- (1) 若有[TO <内存变量>]选项,则将该字符赋给内存变量。
- (2) 若无[<提示信息>],屏幕显示“按任意键继续”。

(3) 该命令的执行与 ACCEPT 基本相同,但只需要输入单个字符,且不需要按 Enter 键。

(4) 一般情况下,提示信息会显示在 Visual FoxPro 主窗口中。如果指定了 Windows 子句,则会出现一个 WAIT 提示窗口,用于显示提示信息。提示窗口一般显示在主窗口的右上角,也可以用 AT 短语指定其在主窗口的位置。

### 3. 键盘输入任意类型数据命令

格式: INPUT [<字符表达式>] TO <内存变量>

功能: 暂停程序的执行,等待用户从键盘上输入数据并赋给内存变量。

说明如下。

(1) 从键盘输入的数据可以是字符型、数值型或逻辑型,也可以是表达式。内存变量的类型由输入的数据类型决定。

(2) 输入字符型数据时必须用界定符界定起来,逻辑型数据须用圆点括起来。

(3) 该命令的其他说明与 ACCEPT 命令一样。

## 5.1.4 顺序结构程序

程序结构是指程序中命令或语句执行的流程结构。Visual FoxPro 系统的程序有两个特点:一是程序控制流模式,由顺序、分支和循环 3 种基本结构构成,每个基本结构可以包含一条或多条语句;二是面向对象可视化的结构程序模块,每个模块是由程式控制流组成的。在 Visual FoxPro 系统的应用程序中,常见的控制结构有顺序结构、选择结构和循环结构。Visual FoxPro 系统提供了相应的语句支持各种结构。

顺序结构是最简单的程序结构,它按命令在程序中出现的先后次序执行。在顺序结构中常用的程序语句有以下 3 种。

### 1. 注释语句

为了增加程序的可读性,常常在程序中写入一些注释、说明性的文字。以“NOTE”或“\*”开头的代码行为注释行。命令行后也可以添加注释,这种注释以“&&”符号开头。注释行为非执行代码,不会影响程序的功能。

### 2. 简单的输出语句

(1) ?表达式 1,表达式 2,...,表达式 n

在新的一行中,顺序输出各表达式的值,如果只有一个“?”号,则可以起到输出换行的作用。

(2) ??表达式 1,表达式 2,...,表达式 n

与?语句作用相似,只是在光标所在处输出。

注意:这里的问号是在半角符号下的问号。在程序中,除引文中的标点符号外,都应在半角环境下输入。



### 3. 命令分行

程序中的每条命令都以 Enter 键结束, 一行只能写一条命令。若命令需要分行书写, 应在一行结束时输入续行符 “;”, 再按 Enter 键。

#### 5.1.5 选择结构程序

支持选择结构的语句包括条件语句和分支语句。

##### 1. 条件语句

语句格式:

IF <条件 1>

<语句序列 1>

[ELSE

<语句序列 2>]

ENDIF

执行过程如图 5-1 所示。

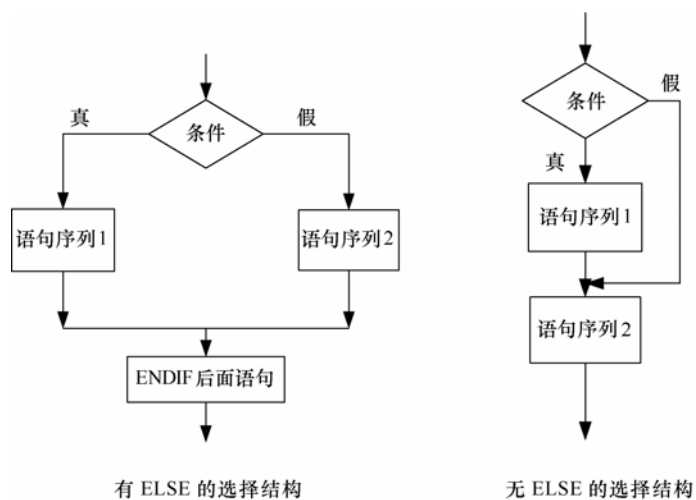


图 5-1 选择结构示意图

多种选择 (IF 语句嵌套)

语句格式:

IF <条件 1>

IF <条件 2>

<命令序列 1>

ELSE

<命令序列 2>

ENDIF

...

```
ELSE
<命令序列 N>
ENDIF
```

## 2. 分支结构

分支结构实现一种扩展的选择结构，它可以根据条件从多组代码中选择一种执行。如图 5-2 所示。

语句格式：

```
DO CASE
 CASE<条件 1>
 <语句序列 1>
 CASE<条件 2>
 <语句序列 2>
 ...
 CASE<条件 n>
 <语句序列 n>
 CASE[OTHERWISE
 <语句序列>]
ENDCASE
```

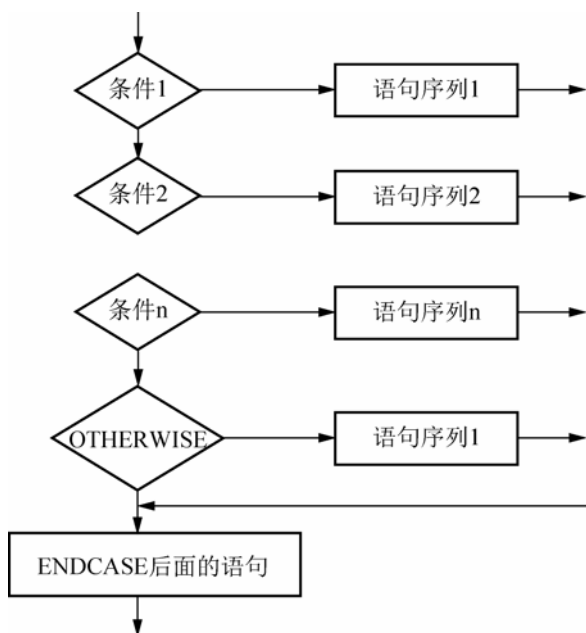


图 5-2 分支结构示意图

### 5.1.6 循环结构程序

循环结构也被称为重复结构，是指在程序执行的过程中，某段代码被重复执行若干次。重复执行的代码段，通常称为循环体。Visual FoxPro 系统支持循环结构的语句包括：DO-WHILE-ENDDO、FOR-ENDFOR 和 SCAN-ENDSCAN。循环结构示意图如图 5-3 所示。

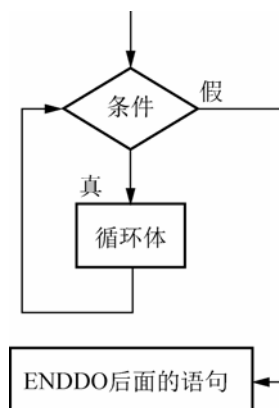


图 5-3 循环结构示意图

### 1. DO WHILE 和 ENDDO 命令

语句格式：

DO WHILE <条件>

<命令序列>

[LOOP]

<命令序列>

[EXIT]

<命令序列>

ENDDO

功能：判断条件是否为真，若为真则重复执行循环体中的<命令序列>，直到<条件>为假，循环结束。

说明如下。

(1) 在使用循环语句时，<条件>的选择是非常重要的，它决定了循环的次数。常用来控制循环的<条件>有以下 4 种。

计数形式。

测试文件开始 (BOF()) 或结束 (EOF()) 函数。

在程序执行中，从键盘上输入数据。

直接用.T.作为条件，这时循环构成“死”循环，但可在循环体中加上退出循环语句 EXIT 来控制循环（常和 IF 语句联用）。

(2) 命令序列中若有[LOOP]项，则跳过该语句到 ENDDO 之间的所有语句，继续进行下一次循环的条件判断。

### 2. SCAN 和 ENDSCAN 命令

与 DO WHILE 和 ENDDO 命令相似，循环可以用 SCAN 和 ENDSCAN 命令实现。

语句格式：

SCAN

<命令序列>

ENDSCAN

功能：在一个表中建立一个执行<命令序列>的循环，并执行对每条记录的操作，直到

表文件记录完为止。

说明：该命令的使用相当于

DO WHILE .NOT. EOF()

<命令序列>

SKIP +1

ENDDO

### 3. FOR-ENDFOR 语句

语句格式：

FOR<循环变量>=<初值> to <终值> [STEP<步长>]

<循环体>

END FOR|NEXT

执行语句时，首先将初值给循环变量，然后判断循环条件是否成立，若循环条件成立，则执行循环体，然后循环变量增加或减小步长值，并再次判断循环条件是否成立，以确定是否再次执行循环体。若循环条件不成立，则结束该循环，执行 ENDFOR 后面的语句。

说明如下。

- (1) <步长>的默认值是 1。
- (2) <初值><终值><步长>都可以是数值表达式。
- (3) 可以在循环体内改变循环变量的值，但这会影响循环体的执行次数。
- (4) EXIT 和 LOOP 命令同样可以出现在该循环体内。

## 5.1.7 过程

### 1. 过程概述

在结构化编程方式中，经常要用到某些具有特定功能的程序段，为了避免重复编写，将这些具有特定功能的程序段组织在一起并存储于一个文件中，其中的每个程序段称为一个过程或子程序。如果一个文件全部由过程组成，那么这种文件就称为过程文件，此时的过程称为外部过程；此外，过程也可以放置在程序文件代码的后面，此时的过程称为内部过程。将一个应用程序划分成一个个功能相对简单、单一的子程序，不仅便于程序开发，也有利于程序的维护。

### 2. 过程文件的建立

命令格式与建立命令文件一样：MODIFY COMMAND <过程文件名>，文件格式同为 .prg。但过程文件内容在格式上要求其中的每个过程的开始必须用下面的语句：PROCEDURE|FUNCTION<过程名>。

过程文件内容如下。

PROCEDURE|FUNCTION <过程名>

<命令组>

[RETURN[<表达式>]]

[ENDPROC|ENDFUNC]

(1) PROCEDURE|FUNCTION 命令表示一个过程的开始，并命名过程。过程名必须以字母或下划线开头，可包含字母、数字和下划线。





(2) ENDPROC|ENDFUNC 命令表示一个过程的结束。如果默认 ENDPROC|ENDFUNC 命令,那么过程结束于下一条 PROCEDURE|FUNCTION 命令或文件结尾处。

(3) 当指令执行到 RETURN 命令时,将转回到调用程序,并返回表达式的值。如果缺省 RETURN 命令,则在过程结束处自动执行一条隐含的 RETURN 命令。若 RETURN 命令不带表达式,则返回逻辑真值 TURE。

过程文件里只包含过程,这些过程可以被任何的其他程序所调用。但在调用过程之前,首先要打开过程文件(内部过程随程序文件的打开而自动打开)。打开过程文件的命令格式为:

SET PROCEDURE TO [<过程文件 1>,<过程文件 2>,...][ADDITIVE]

可以打开一个或多个过程。一旦一个过程文件打开,那么该过程文件中的过程都可以被调用。如果选用 ADDITIVE,那么在打开过程文件时,并不关闭原先已打开的过程文件。

当使用不带任何文件名的 SET PROCEDURE TO 命令时,将关闭所有打开的过程文件。

如果要关闭个别过程文件,可执行以下命令。

RELEASE PROCEDURE <过程文件 1>,<过程文件 2>,...]

存放在命令文件中的过程主要是被命令文件中代码所调用,但也可以被其他程序所调用。

### 3. 过程调用

过程调用格式有两种。

(1) 使用命令。

DO<文件名>|<过程名>

(2) 在名字后面加“()”。

<文件名>|<过程名>()

## 5.1.8 参数传递及变量的作用域

### 1. 用户自定义函数

用户自定义函数是除系统提供的函数外,用户自己定义的某些具有特殊功能的函数,它能增强用户编程的灵活性和通用性。

接收参数的命令有 PARAMETERS 和 LPARAMETERS,格式如下。

PARAMETERS<形参变量 1>,<形参变量 2>,...]

LPARAMETERS<形参变量 1>,<形参变量 2>,...]

不管是 PARAMETERS 还是 LPARAMETERS,都应是子程序的第一条可执行命令。相应的调用格式如下。

格式 1: DO<文件名>|<过程名>WITH<实参>,<实参>,...]

格式 2: <文件名>|<过程名>WITH(<实参>,<实参>,...)]

实参可以是变量、常量,也可以是一般形式的表达式。调用时,系统会自动把实参传递给对应的形参。形参的数目不能少于实参的数目,否则会产生错误。

采用格式 1 调用模块程序时,如果实参是常量或一般形式的表达式,系统会计算出实参的值,并把它们赋值给相应的形参变量,这种情形称为按值传递;如果实参是变量,那么传递的将不是变量的值,而是变量地址,这时形参和实参实际上是同个变量,在模块程序中对形参变量值的改变,同样是对实参变量值的改变,这种情形称为按引用传递。

采用格式 2 调用模块程序时，默认情况下都以按值方式传递参数。如果实参是变量，可以通过命令 SET UPFPARMS 命令重新设置参数传递方式，命令格式如下。

SET UDFPARMS TO VALUE | REFERENCE

TO VALUE：按值传递。

TO REFERENCE：按引用传递。

## 2. 变量的作用域

程序的设计离不开变量。一个变量除了类型和取值之外，还有一个重要的属性就是它的作用域。作用域指出了变量在什么范围内是有效的或能够被访问的。在 Visual FoxPro 中，若以变量的作用域来分，内存变量可分为公共变量、私有变量和局部变量。

### (1) 公共变量。

在任何模块中都可使用的变量称为公共变量。公共变量要先建立后使用，可用 PUBLIC 命令建立。

命令格式：PUBLIC <内存变量表>

并为它们赋值逻辑假.F。

公共变量一旦建立就有效，即使程序运行结束返回到命令窗口也不会消失。只有当执行 CLEAR MEMORY、RELEASE、QUIT 等命令后，公共变量才被释放。

### (2) 私有变量。

在程序中直接使用的由系统自动隐含建立的变量都是私有变量。私有变量的作用域建立在它的模块及其下属的各层模块中。一旦建立它的模块程序运行结束，这些私有变量将自动清除。

### (3) 局部变量。

局部变量只能在建立它的模块中使用，不能在上层或下层模块中用。当建立它的模块程序运行结束时，局部变量将自动释放。

命令格式：LOCAL <内存变量表>

该命令建立指定的局部内存变量，并为它们赋初值假.F。

局部变量要先建立，后使用。

## 任务 5.2 实例训练

### 5.2.1 顺序结构程序设计实例

【例 5-1】查看下面程序段的运行结果。（程序 5-1）

程序代码如下。

```
x=16
? "x 的平方根为: ",sqrt(x)
? "x 的绝对值为: ",abs(x)
s1="beijing"
s2="good"
? "s1=Beijing"
? "s2=good"
```



```

? "substr(s1,2,3) ",substr(s1,2,3)
? "字符串 s1 的长度为: ",len(s1)
? "s1+s2=",s1+s2
? "今天的日期为: ",date()
? "现在的时间为: ",time()

```

程序 5-1 的运行结果如图 5-4 所示。

```

x 的平方根为: 4.00
x 的绝对值为: 16
s1=Beijing
s2=good
substr(s1,2,3) eij
字符串s1 的长度为: 7
s1+s2= beijinggood
今天的日期为: 03/18/11
现在的时间为: 09:54:14

```

图 5-4 程序 5-1 的运行结果

### 5.2.2 选择结构程序设计实例

【例 5-2】某地到上海的里程为 1 000km，通过运输公司向上海运输物资，计费标准为 500 元/t，但超过 5t，超出数的计费标准为 400 元/t。试编写程序计算运费。（程序 5-2）

程序代码：

```

INPUT "请输入物资重量：" TO x
IF x<=5
Y=x*500
ELSE
Y=5*500+(x-5)*450
ENDIF
?"运费为：" ,Y

```

```

请输入物资重量: 6
运费为: 2950

```

图 5-5 程序 5-2 的运行结果

按照上面讲述的运行方式运行该程序，在运行界面上会出现“请输入物资重量：”，假设我们输入 6，然后按 Enter 键，则会出现“运费为：2950”。程序 5-2 的运行结果如图 5-5 所示。

【例 5-3】首先建立一个表 BOOK.DBF，表中记录如图 5-6 所示，然后在 BOOK.DBF 表文件中查找《计算机概论》的库存量，若少于 300 本，则需要购书，若大于 300 本，则需要调书到其他部门，使书表库存量为 300 本。要求打印出购书或调出书的册数，并修改书表的“库存”和“总金额”字段的内容。（程序 5-3）

| 图书编号 | 书名      | 库存  | 定价      | 总金额       |
|------|---------|-----|---------|-----------|
| 001  | 计算机概论   | 400 | 20.0000 | 8000.0000 |
| 002  | 数据结构    | 200 | 30.0000 | 6000.0000 |
| 003  | VFP程序设计 | 320 | 25.0000 | 8000.0000 |

图 5-6 BOOK 表初始数据

程序代码如下。

```
SET TALK OFF
USE BOOK
LOCATE ALL FOR 书名="计算机概论"
DISPLAY
A=库存
IF A<300
 B=300-A
 ? "应购书:",B,"册"
 REPLACE 库存 WITH 库存+B,总金额 WITH 300*定价
ELSE
 IF A>300
 B=A-300
 REPLACE 库存 WITH 库存-B,总金额 WITH 300*定价
 ? "应调出书:",B,"册"
 ELSE
 ? "该书表库存正常"
 ENDIF
ENDIF
LIST
USE
SET TALK ON
```

程序 5-3 的运行结果如图 5-7 所示。

| 记录号         | 图书编号 | 书名      | 库存  | 定价      | 总金额       |
|-------------|------|---------|-----|---------|-----------|
| 1           | 001  | 计算机概论   | 400 | 20.0000 | 8000.0000 |
| 应调出书: 100 册 |      |         |     |         |           |
| 记录号         | 图书编号 | 书名      | 库存  | 定价      | 总金额       |
| 1           | 001  | 计算机概论   | 300 | 20.0000 | 6000.0000 |
| 2           | 002  | 数据结构    | 200 | 30.0000 | 6000.0000 |
| 3           | 003  | VFP程序设计 | 320 | 25.0000 | 8000.0000 |

图 5-7 程序 5-3 的运行结果

此时我们再观察 BOOK 表中的记录，如图 5-8 所示。



| 图书编号 | 书名      | 库存  | 定价      | 总金额       |
|------|---------|-----|---------|-----------|
| 001  | 计算机概论   | 300 | 20.0000 | 6000.0000 |
| 002  | 数据结构    | 200 | 30.0000 | 6000.0000 |
| 003  | VFP程序设计 | 320 | 25.0000 | 8000.0000 |

图 5-8 程序运行后 BOOK 表中记录

用 IF-ELSE-ENDIF 进行多次嵌套时很容易出错，系统给出了下面的命令，能使程序更直观和清楚。

【例 5-4】假设奖金  $jj$  与税率  $r$  之间存在着如下对应关系。

|                    |             |  |         |
|--------------------|-------------|--|---------|
| $jj < 2000$        |             |  | $r=0\%$ |
| $2000 < jj < 3000$ | 超过 2000 的部分 |  | $r=5\%$ |
| $3000 < jj < 5000$ | 超过 3000 的部分 |  | $r=8\%$ |



5000<jj 10000      超过 5000 的部分      r=10%  
 jj>10000      超过 10000 的部分      r=15%

设计一个程序 5-4, 给出你的奖金, 求出应缴纳的税金。

程序代码如下。

```
SET TALK OFF
input "请输入你的奖金：" to jj
jds1=50 &&奖金为 3000 元时应缴纳的税金
jds2=50+160 &&奖金为 5000 元时应缴纳的税金
jds3=50+160+500 &&奖金为 10000 元时应缴纳的税金
DO case
CASE jj<=2000
sj=0
CASE jj<=3000
sj=(jj-2000)*0.05
CASE jj<=5000
sj=jds1+(jj-3000)*0.08
CASE jj<=10000
sj=jds2+(jj-5000)*0.1
OTHERWISE
sj=jds3+(jj-10000)*0.15
ENDCASE
?"你的税金是:",sj
SET TALK ON
```

程序 5-4 的运行结果如图 5-9 所示。

|               |       |
|---------------|-------|
| 请输入你的奖金: 2300 |       |
| 你的税金是:        | 15.00 |

图 5-9 程序 5-4 的运行结果

### 5.2.3 循环结构程序设计实例

【例 5-5】计算  $S=1+2+\dots+100$  的值。(程序 5-5)

该程序要使用循环结构, 解题的思路可归纳为以下两点。

(1) 引进变量 S 和 I, S 用来保存累加的结果, 初值为 0; I 既作为被累加的数据, 也作为控制循环条件是否成立的变量, 初值为 1。

(2) 重复执行命令  $S=S+I$  和  $I=I+1$ , 直到 I 的值超过 100。每次执行, S 的值增加 I, I 的值增加 1。

程序代码如下。

```
S=0
I=1
DO WHILE I<=100
```

```

S=S+I
I=I+1
ENDDO
?'S 的值为：',S

```

程序 5-5 的运行结果如图 5-10 所示。



图 5-10 程序 5-5 的运行结果

**【例 5-6】**显示 students.dbf 表中每个同学的姓名、性别、入学成绩和籍贯。(程序 5-6)  
程序代码如下。

```

SET TALK OFF
CLEAR
USE STUDENTS
SCAN
?'姓名：',姓名
?'性别：',性别
?'入学成绩：',入学成绩
?'籍贯：',籍贯
ENDSCAN
USE
SET TALK ON

```

程序 5-6 的运行结果如图 5-11 所示。

**【例 5-7】**从键盘上输入 10 个数，找出其中的最大数与最小数。(程序 5-7)  
程序代码如下。

```

CLEAR
INPUT "请输入一个数：" TO x
STORE x TO max,min
FOR i=2 TO 10
INPUT "请输入一个数：" TO x
IF max<x
max=x
ENDIF
IF min>x
min=x
ENDIF
ENDFOR
?'最大值：',max
?'最小值：',min
RETURN

```

程序 5-7 的运行结果如图 5-12 所示。



|       |      |
|-------|------|
| 姓名:   | 陈达文  |
| 性别:   | 男    |
| 入学成绩: | 345  |
| 籍贯:   | 新疆奎屯 |
| 姓名:   | 陈桂芬  |
| 性别:   | 女    |
| 入学成绩: | 240  |
| 籍贯:   | 新疆伊犁 |
| 姓名:   | 江培强  |
| 性别:   | 男    |
| 入学成绩: | 420  |
| 籍贯:   | 广东高州 |
| 姓名:   | 林江文  |
| 性别:   | 男    |
| 入学成绩: | 380  |
| 籍贯:   | 广东湛江 |
| 姓名:   | 林世鑫  |
| 性别:   | 男    |
| 入学成绩: | 290  |
| 籍贯:   | 广东徐闻 |
| 姓名:   | 陈玉强  |
| 性别:   | 男    |
| 入学成绩: | 315  |
| 籍贯:   | 广东湛江 |
| 姓名:   | 谭旭金  |
| 性别:   | 男    |
| 入学成绩: | 300  |
| 籍贯:   | 广东廉江 |
| 姓名:   | 洪明信  |
| 性别:   | 男    |
| 入学成绩: | 320  |
| 籍贯:   | 广东信宜 |
| 姓名:   | 李小华  |
| 性别:   | 女    |
| 入学成绩: | 310  |
| 籍贯:   | 广东揭阳 |
| 姓名:   | 张晶林  |
| 性别:   | 男    |
| 入学成绩: | 310  |
| 籍贯:   | 河南郑州 |

图 5-11 程序 5-6 的运行结果

|         |   |
|---------|---|
| 请输入一个数: | 1 |
| 请输入一个数: | 2 |
| 请输入一个数: | 3 |
| 请输入一个数: | 4 |
| 请输入一个数: | 5 |
| 请输入一个数: | 6 |
| 请输入一个数: | 7 |
| 请输入一个数: | 8 |
| 请输入一个数: | 9 |
| 请输入一个数: | 0 |
| 最大值:    | 9 |
| 最小值:    | 0 |

图 5-12 程序 5-7 的运行结果

#### 5.2.4 过程调用实例

【例 5-8】程序中常常需要计算阶乘、圆面积和长方形面积，现将它们编写为 3 个过程，放在一个过程文件（GC.PRG）中，被主程序（ZC.PRG）调用。（程序 5-8）

程序代码如下。

(1) ZC.PRG 代码。

```

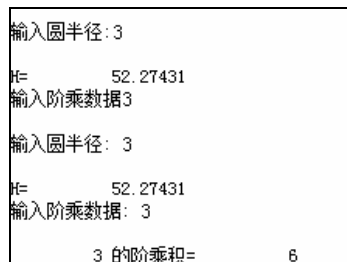
PUBLIC S,T
SET PROCEDURE TO GC
INPUT "输入圆半径: " TO R
DO P1
H=S
DO P2 WITH 4,6
H=H+S
? "H=",H
INPUT "输入阶乘数据: " TO I
DO P3
CLOSE PROCEDURE

```

(2) GC.PRG 代码。

```
PROCEDURE P1
S=3.14159*R*R
RETURN
PROCEDURE P2
PARAMETERS 长,宽
面积=长*宽
S=面积
RETURN
PROCEDURE P3
N=1
T=1
DO WHILE N<=I
T=T*N
N=N+1
ENDDO
DO P4
RETURN
PROCEDURE P4
? I, "的阶乘积=",T
RETURN
```

程序 5-8 的运行结果如图 5-13 所示。



```
输入圆半径: 3
H= 52.27431
输入阶乘数据: 3
3 的阶乘积= 6
```

图 5-13 程序 5-8 的运行结果

### 5.2.5 参数传递实例

【例 5-9】按值传递和按引用传递示例。(程序 5-9)

程序代码如下。

```
CLEAR
STORE 100 TO X1,X2
DO PRO WITH X1,X2 &&按引用传递
?"第一次:",X1,X2
STORE 100 TO X1,X2
DO PRO WITH (X1),(X2) &&按值传递
?"第二次:",X1,X2
STORE 100 TO X1,X2
DO PRO WITH X1+1,X2+1 &&按值传递
```





```

? "第三次: ", X1, X2
STORE 100 TO X1, X2
 PRO(X1, X2) &&按值传递
? "第四次: ", X1, X2
STORE 100 TO X1, X2
 PRO(@X1, @X2) &&按引用传递
? "第五次: ", X1, X2
*过程 pro
PROCEDURE PRO
PARAMETERS X1, X2
X1=X1+1
X2=X2+1
ENDPROC

```

程序 5-9 的运行结果如图 5-14 所示。

|      |     |     |
|------|-----|-----|
| 第一次: | 101 | 101 |
| 第二次: | 100 | 100 |
| 第三次: | 100 | 100 |
| 第四次: | 100 | 100 |
| 第五次: | 101 | 101 |

图 5-14 程序 5-9 的运行结果

### 5.2.6 变量的作用域实例

【例 5-10】写出以下程序的运行结果。(程序 5-10)

程序代码如下。

```

PUBLIC x1
LOCAL x2
STORE "F " TO x3
DO p6
? "主程序中..."
? "x1= ", x1
? "x2= ", x2
? "x3= ", x3
RETURN
PROCEDURE p6
? "子程序中..."
? "x1= ", x1
? "x3= ", x3
RETURN

```

程序 5-10 的运行结果如图 5-15 所示。

开发应用程序时，主程序和子程序不一定是由同一个人来设计的。编写子程序的人不可能对主程序的变量了解得十分清楚。这样就出现了以下情况：子程序中用到的变量实际上在主程序中已经建立，子程序的运行会无意间改变主程序中变量的取值。为解决问题，可在子程序中使用 PRIVATE 命令隐藏主程序中可能存在的变量，使得这些变量在子程序中暂时无效。

```

子程序中...
x1= .F.
x3= F
主程序中...
x1= .F.
x2= .F.
x3= F
返回命令窗口时...
x1= .F.

```

图 5-15 程序 5-10 的运行结果

命令格式：

PRIVATE<内存变量表>

PRIVATE ALL [LIKE<通配符>|EXCEPT<通配符>]

它的作用就是隐藏指定的在上层模块中可能已经存在的内存变量，使得这些变量在当前模块程序中暂时无效。

【例 5-11】写出以下程序的运行结果。（程序 5-11）

程序代码如下。

```

SET TALK OFF
val1=10
val2=15
DO P7
? val1,val2
PROCEDURE P7
PRIVATE val1
val1=50
val2=100
? val1,val2
RETURN

```

程序 5-11 的运行结果如图 5-16 所示。

|    |     |
|----|-----|
| 50 | 100 |
| 10 | 100 |

图 5-16 程序 5-11 的运行结果

## 任务 5.3 项目分析

### 5.3.1 查询程序

编写一段顺序结构的程序，名为“查询程序.prg”，实现以下功能：在数据库表“学生信息表 students”中查找“学号”字段为 0041001 的学生的情况。

程序代码如下。

```

SET TALK off
CLEAR

```



```
USE students
LOCATE ALL FOR 学号="0041001"
display
SET TALK on
```

“查询程序”运行结果如图 5-17 所示。

| 记录号 | 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   | 照片  | 备注 |
|-----|---------|-----|---------|----|----|------|------|-----|----|
| 8   | 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 | gen | 备注 |

图 5-17 “查询程序”运行结果

### 5.3.2 成绩等级评定程序

编写一段程序，名为“等级评定.prg”，对选择 jsj001 这门课的学生进行成绩评定。评定方式如下。

优：jsj001 成绩 $\geq 90$

良：jsj001 成绩 $\geq 80$

中：jsj001 成绩 $\geq 60$

差：jsj001 成绩 $< 60$

程序代码如下。

```
CLEAR
SET TALK OFF
USE SCORES
SCAN
IF 课程编号="jsj001"
DO CASE
CASE 成绩<60
DJ="差"
CASE 成绩>=60 .AND. 成绩<80
DJ="中"
CASE 成绩>=80 .AND. 成绩<90
DJ="良"
CASE 成绩>=90 .AND. 成绩<100
DJ="优"
OTHERWISE
?"成绩输入错误!"
ENDCASE
?"学号：",学号,"等级：",DJ
ELSE
ENDIF
ENDSCAN
SET TALK ON
```

“等级评定”程序运行结果如图 5-18 所示。

|     |         |     |   |
|-----|---------|-----|---|
| 学号: | 0104001 | 等级: | 良 |
| 学号: | 0104005 | 等级: | 良 |
| 学号: | 0104002 | 等级: | 中 |
| 学号: | 0104003 | 等级: | 优 |
| 学号: | 0104004 | 等级: | 中 |

图 5-18 “等级评定”程序运行结果

### 5.3.3 修改成绩程序

编写一段程序，名为“修改成绩.prg”，实现以下功能：对“成绩表 scores”中的“成绩”做出修改，若成绩 $\geq 75$ ，则提高 10%；若成绩 $< 75$ ，则提高 5%。

程序代码如下。

```
CLEAR
SET TALK OFF
USE SCORES
SCAN
A=成绩
IF A>=75
 B=A*1.1
 REPLACE 成绩 WITH B
ELSE
 B=A*1.05
 REPLACE 成绩 WITH B
ENDIF
ENDSCAN
SET TALK ON
```

## 任务 5.4 模仿设计与自主设计

### 5.4.1 闰年程序

编写一段程序，名为“闰年程序.prg”，实现如下功能：从键盘任意输入一个年份，输出其是否是闰年的信息。

### 5.4.2 随机数产生程序

编写一段程序，名为“随机产生整数程序.prg”，实现如下功能：随机产生一个 10 以内的整数。

### 5.4.3 素数程序

编写一段程序，打印出 3 ~ 1 000 以内的所有素数。要求将求素数的过程写在一个过程文件中。



## 项目小结 5

项目 5 介绍了 Visual FoxPro 程序设计基础,包括程序的建立和编辑方法,简单的输入/输出语句,过程化程序设计基本结构,以及过程文件的建立及其调用方法。

## 项目习题 5

### 1. 选择题

- (1) 在 Visual FoxPro 中,关于过程调用的叙述正确的是 ( )。
- A. 当实参的数量少于形参的数量时,多余的形参初值取逻辑假
  - B. 当实参的数量多于形参的数量时,多余的实参被忽略
  - C. 实参与形参的数量必须相等
  - D. A 和 B 都正确
- (2) 在 Visual FoxPro 中,如果希望一个内存变量只限于在本过程中使用,说明这种内存变量的命令是 ( )。
- A. PRIVATE      B. PUBLIC
  - C. LOCAL      D. 在程序中直接使用的内存变量(不通过 A、B、C 说明)
- (3) 在 DO WHILE-ENDDO 循环结构中,LOOP 命令的作用是 ( )。
- A. 退出过程,返回程序开始处
  - B. 转移到 DO WHILE 语句行,开始下一个判断和循环
  - C. 终止循环,将控制转移到本循环结构 ENDDO 后面的第一条语句继续执行
  - D. 终止程序执行
- (4) 下列程序段的输出结果是 ( )。

```
Clear
Store 10 to a
Store 20 to b
Set udfparms to reference
Do swap with a,(B)
?a,b
Procedure swap
Parameters x1,x2
temp=x1
x1=x2
x2=temp
endproc
```

- A. 10 20      B. 20 20      C. 20 10      D. 10 10

(5) 使用调试器调试一个程序, 如果想在过程执行时观察 X1 的值, 可以在其中安置一条命令, 程序执行到该命令时, 系统将计算 X1 的值, 并将结果在调试输出窗口中显示, 这条命令的正确写法是 ( )。

- A. DEBUGOUT X1                      B. DEBUG X1  
C. OUT X1                              D. TEST X1

(6) 在 Visual FoxPro 中, 如果希望跳出 SCAN-ENDSCAN 循环, 执行 ENDSCAN 后面的命令, 应该使用 ( )。

- A. LOOP                                  B. EXIT  
C. BREAK                                D. RETURN

(7) 执行如下程序后, 最后一条命令的显示结果是 ( )。

```
DIMENSION M(2,2)
M(1,1)=10
M(1,2)=20
M(2,1)=30
M(2,2)=40
? M(2)
```

- A. 变量未定义的提示                      B. 10  
C. 20                                      D. .F.

(8) 下列叙述中正确的是 ( )。

- A. 软件测试应该由程序开发者来完成  
B. 程序经调试后一般不需要再测试  
C. 软件维护只包括对程序代码的维护  
D. 以上三种说法都不对

(9) 在 Visual FoxPro 中, 用于调用模块程序的命令是 ( )。

- A. PROCEDURE<过程名>                      B. SET PROCEDURE TO <过程文件>  
C. FUNCTION<过程名>                      D. DO<过程名>|<过程名> WAIT<实参>

(10) 执行如下程序, 如果输入 N 值为 6, 则最后 S、I 的显示结果是 ( )。

```
SET TALK OFF
S=0
I=1
INPUT "N=" TO N
DO WHILE S<=N
 S=S+I
 I=I+1
ENDDO
? S, I
SET TALK ON
```

- A. 5 2                                      B. 6 3                                      C. 10 5                                      D. 12 6

## 2. 填空题

(1) 在 Visual FoxPro 中, 以下程序的运行结果 (即执行命令 DO main 后) 是\_\_\_\_\_。



\*程序文件名：main.prg

```
SET TALK OFF
CLOSE ALL
CLEAR ALL
mX="Visual Foxpro"
mY="二级"
DO s1
?mY+mX
RETURN
*子程序文件名：s1.prg
PROCEDURE s1
LOCAL mX
mX="Visual FoxPro DBMS 考试"
mY="计算机等级"+mY
RETURN
```

(2) 以下程序显示的结果是\_\_\_\_\_。

```
s=1
I=0
Do while I<8
 s=s+I
 I=I+2
Enddo
? s
```

(3) 程序测试分为静态分析和动态测试。其中\_\_\_\_\_是指不执行程序，而只对程序文本进行检查，通过阅读和讨论，分析和发现程序中的错误。

(4) 在 Visual FoxPro 中，建立或修改过程文件的命令是\_\_\_\_\_，文件默认的扩展名是\_\_\_\_\_。

(5) 如果一个过程不包含 RETURN 语句，或者 RETURN 中没有指定表达式，那么该过程返回\_\_\_\_\_值。

(6) 在 Visual FoxPro 中，将只能在建立它的模块中使用的内存变量称为\_\_\_\_\_。



# 项目 6

## 设计学生学籍管理 系统的表单

### 项目要点◎

表单的建立与操作。  
常用表单控件及面向对象的概念。

### 学习目标◎

了解表单的基本知识、面向对象的基本概念。  
掌握表单设计器的功能、表单的建立和运行方法。  
理解掌握表单控件的类型及设置表单控件属性的基本方法。

Visual FoxPro 的表单 (Form) 在有的书上叫“窗体”，是用户常用的查看数据库和输入数据的屏幕“界面”。它提供了丰富的能够反映用户事件的对象集，让用户方便地完成信息管理的任务，可见表单远非只是“界面”那么简单。

表单为数据库信息的显示、输入和编辑提供了非常简便的方法，表单的设计是可视化编程的基础。Visual FoxPro 提供了一个非常强的表单设计器 (Form Designer)，它是可视化的面向对象结构的工具。本项目将介绍在 Visual FoxPro 环境下建立、修改、修饰、运行表单，以及向表单中添加控件、设计表程序模块等内容。

### 任务 6.1 熟悉表单基础知识

#### 6.1.1 表单设计器环境

本节主要介绍表单设计器的环境，以及如何添加表单控件、管理表单控件和设置表单





数据环境。整个表单设计界面如图 6-1 所示。

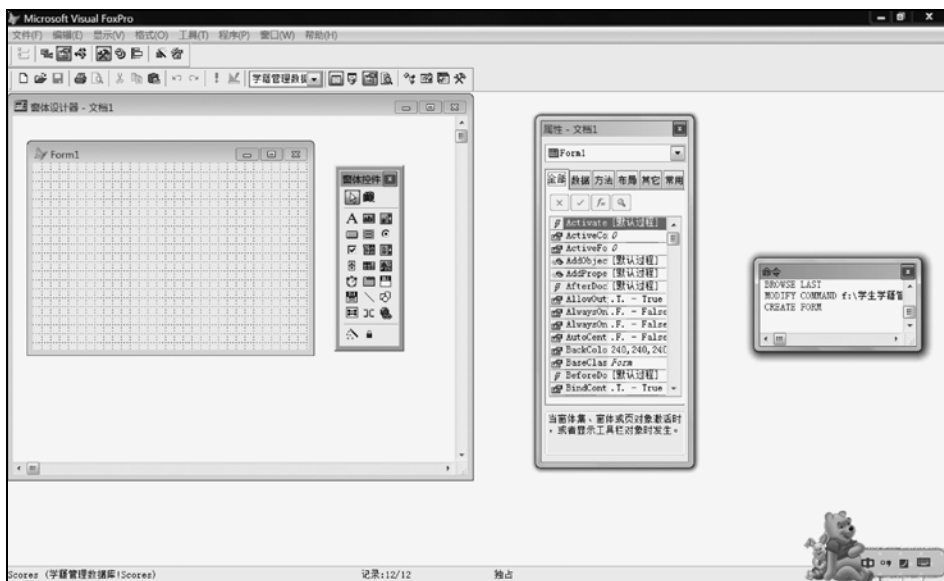


图 6-1 表单设计界面



注意：本书图片中出现的“窗体设计器”、“窗体控件”等与“表单设计器”、“表单控件”等是一样的，在后面的讲述中，我们将统一使用“表单设计器”、“表单控件”等作为文字叙述。



图 6-2 属性窗口

### 1. 表单设计器窗口

用户可以在表单设计器窗口中添加控件。

### 2. 属性窗口

属性窗口可以通过单击“表单设计器”工具栏上的“属性窗口”按钮来显示，如图 6-2 所示。


该窗口中有对象框，属性列表框，属性、方法和事件列表框。对象框显示的是当前选定的对象名称，单击对象框右侧的 ▾ 按钮，将打开所有对象的名称列表，用户可以选择一个需要编辑修改的对象或表单。

属性窗口中的列表框显示当前被选定对象的所有属性、方法与事件，用户也可以从中选择一个。如果选择的是属性项，窗口内将出现属性设置框，用户可以在此对选定的属性进行设置。

对于表单及控件的绝大多数属性，其数据类型通常是固定的。如 Width 属性只能是数据型，Caption 属性只能接收字符型数据。但有些属性可以是任意数据类型，复选框的 Value 属性可以是数值型，

也可以是逻辑型。一般说来,要设置属性为字符型,可在设置框中直接输入,不需要加定界符。但对那些既可以接收数值型数据又可接收字符型数据的属性,要为其加定界符。如输入“123”,默认为数值型,若要设置为字符串,则要用"123"来表示。

在属性设置框中,也可输入表达式,不过要先输入“=”号,或者利用表达式生成器来给属性生成一个表达式。表达式在对象初始化时计算。

有些属性要从系统提供的一组属性中指定,此时可单击设置属性窗口中的  按钮打开列表框选择。

要将一个属性设置为默认,可在属性列表中右击该属性,从快捷菜单中选择“重置为默认值”选项。要将一个属性置为空串,可在选中后,依次按 Backspace 键和回车键,此时属性列表中显示“无”。也可以同时选择多个对象,这时属性窗口显示这些对象的共有属性,用户对属性的设置也将针对被选定的对象。

表单设计中常用的属性、事件与方法如表 6-1 所示。

表 6-1 表单设计中常用的属性、事件与方法

| 属性、事件和方法       | 说 明                                                                  | 默 认 值       |
|----------------|----------------------------------------------------------------------|-------------|
| AlwaysOnTop 属性 | 控制表单是否总是处在其他打开窗口之上                                                   | 假 (.F.)     |
| AutoCenter 属性  | 控制表单初始化时是否让表单自动地在 Visual FoxPro 主窗口中居中                               | 假 (.F.)     |
| BackColor 属性   | 决定表单窗口的颜色                                                            | 255,255,255 |
| BorderStyle 属性 | 决定表单是否有边框,若有边框,是单线边框、双线边框,还是系统边框。如果 BorderStyle 为 3 (系统),用户可重新改变表单大小 | 3           |
| Caption 属性     | 决定表单标题栏显示的文本                                                         | Form1       |
| Closable 属性    | 控制用户是否能通过双击“关闭”框来关闭表单                                                | 真 (.T.)     |
| MaxButton 属性   | 控制表单是否具有最大化按钮                                                        | 真 (.T.)     |
| MinButton 属性   | 控制表单是否具有最小化按钮                                                        | 真 (.T.)     |
| Movable 属性     | 控制表单是否能移动到屏幕的新位置                                                     | 真 (.T.)     |
| WindowState 属性 | 控制表单是最小化、最大化还是正常状态                                                   | 0 正常        |
| WindowType 属性  | 控制表单是非模式表单(默认)还是模式表单。如果是模式表单,用户在访问应用程序用户界面中任何其他单元前必须关闭该表单            | 0 非模式       |
| Activate 事件    | 当激活表单时发生                                                             |             |
| Click 事件       | 在控件上单击鼠标左键时发生                                                        |             |
| DblClick 事件    | 在控件上双击鼠标左键时发生                                                        |             |
| Destroy 事件     | 当释放一个对象的实例时发生                                                        |             |
| Init 事件        | 在创建表单对象时发生                                                           |             |
| Error 事件       | 当某方法(过程)运行出错时发生                                                      |             |
| KeyPress 事件    | 当按下并释放某个键时发生                                                         |             |
| Load 事件        | 在创建表单对象前发生                                                           |             |
| Unload 事件      | 当对象释放时发生                                                             |             |
| RightClick 事件  | 在单击鼠标右键时发生                                                           |             |
| AddObject 方法   | 运行时,在容器对象中添加对象                                                       |             |
| Move 方法        | 移动一个对象                                                               |             |
| Refresh 方法     | 重画表单或控制,并刷新所有值                                                       |             |
| Release 方法     | 从内存中释放表单                                                             |             |
| Show 方法        | 显示一张表单                                                               |             |

### 3. “表单控件”工具栏

“表单控件”工具栏,如图 6-3 所示,该工具栏用于向表单窗口添加控件。方法是先单

击“表单控件”工具栏中的控件，然后在表单窗口中合适的位置单击或拖动鼠标来确定控件的大小。

除了控件按钮，“表单控件”工具栏还有 4 个辅助按钮。

“选择对象”按钮：当该按钮处于按下状态时，表示不可创建控件；当该按钮处于凸起状态时，表示允许创建控件。

“按钮锁定”按钮：当该按钮处于按下状态时，可从“表单控件”工具栏中单击某控件按钮，然后在表单窗口中连续添加这种类型的多个控件。

“生成器锁定”按钮：当该按钮处于按下状态时，每次添加控件，系统都会打开相应的生成器对话框。

“查看类”按钮：在可视化设计菜单中，除了可以使用系统提供的一些基本类，还可以使用保存在类库中的用户自定义类，但应先将它们添加到“表单控件”工具栏中。将一个类添加到“表单控件”工具栏的方法是单击“表单控件”工具栏中的“查看类”按钮，在弹出的下拉菜单中选择“添加”命令，弹出“打开”对话框，在该对话框中选定所需要的类库文件，并单击“确定”按钮。

#### 4. “表单设计器”工具栏

“表单设计器”工具栏如图 6-4 所示。



图 6-3 “表单控件”工具栏

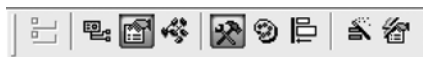


图 6-4 “表单设计器”工具栏

该工具栏内含“设置 Tab 键次序”、“数据环境”、“属性窗口”、“代码窗口”、“表单控件工具栏”、“调色板工具栏”、“布局工具栏”、“表单生成器”和“自动格式”按钮。

#### 5. “表单”菜单

“表单”菜单中的命令主要用于创建、编辑表单或表单集。

##### 6.1.2 表单控件的基本操作

在表单设计器环境下，经常需要对表单上的控件进行移动、复制和删除等基本操作。

##### 1. 选定控件

用鼠标单击即可选定该控件，被选定的控件周围有 8 个控点。也可以同时选定多个控件，如果是相邻的控件，只需要将“表单控件”工具栏上的“选定”按钮按下，同时拖动

鼠标框选控件即可。如果要选定多个不相邻的控件，可以在按住 Shift 键的同时，依次单击各控件。

## 2. 移动控件

先选定控件，然后用鼠标将控件拖动到需要的位置上。如果在拖动鼠标的同时按住 Ctrl 键，可以使鼠标的移动步长减小，使用方向键也可以移动控件。

## 3. 调整控件的大小

选定控件，然后拖动控件四周的某个点来改变控件的大小。

## 4. 复制控件

先选定控件，接着选择“编辑”菜单中的“复制”与“粘贴”命令。

## 5. 删除控件

选定要删除的控件，然后按 Delete 键进行删除。

### 6.1.3 常用表单控件

表单的设计离不开控件，要很好地设计和使用控件，需要了解控件的属性、方法和事件。本节主要介绍几种常用的控件。

#### 1. 标签控件（Label）

标签控件是按一定格式显示在表单上的文本信息，用来显示表单中各种说明和提示。一旦标签控件的属性、事件和方法被定义，输出信息将根据这些定义，按指定的风格输出。

标签的主要属性有：标签的大小、颜色，显示信息的内容、字体大小和风格等。

#### 2. 命令按钮控件

命令按钮控件主要是用来控制程序的执行过程与数据表中数据的操作的。

在设计系统时，程序设计要经常添加具有不同功能的命令按钮，提供不同的操作。只要将不同操作的代码存入不同命令按钮的 Click 事件中，当表单运行时，用户单击某一按钮，就将触发该命令按钮的 Click 事件代码完成指定的操作。

#### 3. 文本框控件

文本框控件主要用于数据表中某些字段（非备注型字段和通用型字段）的输入、输出，以及从窗口为内存变量赋值等操作。

它与标签控件最大的不同是使用的数据源不同。标签控件的数据源来自其本身的属性 Caption，而文本框控件数据源来自数据表中非备注型字段、通用型字段的其他字段和内存变量。

文本框控件的主要属性包括：文本框的大小，文本框中输入与输出信息的字体大小、风格和颜色等。



#### 4. 编辑框控件

与文本框控件一样，编辑框控件也是用来输入、编辑数据的，但它有自己的特点。

(1) 编辑框控件实际上是一个完整的字处理器，利用它能选择、剪切、粘贴和复制正文，可以实现自动换行，能够有自己的垂直滚动条，可以在正文里移动光标。

(2) 编辑框只能输入、编辑字符型数据，包括字符型内存变量、数组元素、字段及备注字段的内容。

#### 5. 复选框控件

复选框主要用于标记一个两值状态，如真 (.T.) 或假 (.F.)。当处于“真”状态时，复选框内显示一个对勾；否则，复选框内为空白。

其主要属性如下。

(1) Value 属性，用来指明复选框的当前状态。复选框的 Value 属性值有 3 种情况，如表 6-2 所示。

(2) ControlSource 属性，指明与复选框建立联系的数据源。

表 6-2 复选框的 Value 属性值

| 属 性 值     | 说 明         |
|-----------|-------------|
| 0 或.F.    | 未被选中 (默认)   |
| 1 或.T.    | 被选中         |
| 2 或.null. | 不确定，只在代码中有效 |

#### 6. 选项组控件

选项组控件又称为单选按钮组，是包含单选按钮的一种容器。一个选项组控件中往往有若干个选项按钮，但用户只能从中选择一个按钮，而选项组中的其他按钮，不管原来是什么状态，都变为未选中状态。被选中的单选按钮内会显示一个圆点。

其主要属性如下。

(1) ButtonCount 属性，指定一个选项按钮组中的按钮数目。

(2) Value 属性，指定控件的当前状态。

(3) ControlSource 属性，指定与对象建立联系的数据源。

(4) Buttons 属性，用于存取一个组中每个按钮的数目。

#### 7. 列表框控件

列表框提供一组条目，用户可以从选择一个或多个条目。一般情况下，列表框显示其中的若干条目，用户可以通过滚动条来浏览其他条目。

其主要属性如下。

(1) RowsSourceType 属性与 Rowsource 属性。RowsSourceType 指明列表框中条目数据源的类型，Rowsource 指定列表框的条目数据源。

(2) List 属性，用于存取列表框中数据条目的字符串数组。

(3) ListCount 属性，指明列表框中数据条目的数目。

(4) ColumnCount 属性，指定列表框的列数。

(5) Value 属性, 返回列表框中被选中的条目数。

(6) ControlSource 属性, 指定一个字段或变量的取值及类型保持一致。

(7) Selected 属性, 指定列表框中的某个条目是否处于选定状态。

(8) MultiSelect 属性, 指定用户能否在列表框控件内进行多重选定。如果是 0 或.F., 则不允许多重选择; 如果是 1 或.T., 则允许多重选择。要选择多个条目, 可按住 Ctrl 键并用鼠标单击条目。

## 8. 组合框控件

与列表框控件类似, 组合框也提供一组条目供用户从中选择。它们的主要区别如下。

(1) 对于组合框控件, 通常只有一个条目可见, 所以相比列表框控件, 组合框控件能节省表单中的显示空间。

(2) 组合框控件不提供多重选择功能, 没有 Multiselect 属性。

(3) 组合框控件有两种形式, 即下拉列表框与列表框。

如图 6-5 所示的组合框控件, 一个为下拉列表框控件, 一个为列表框控件。

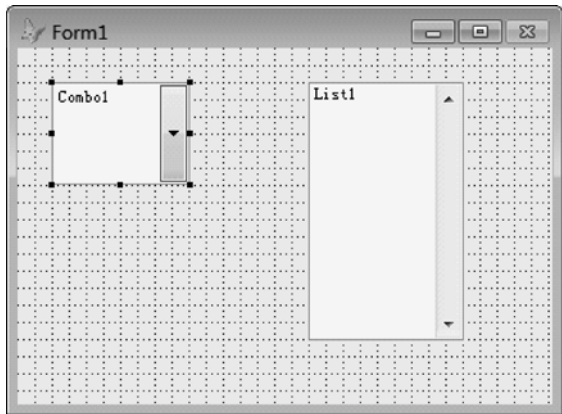


图 6-5 组合框控件

## 9. 表格 (grid) 控件

表格是一种容器对象, 其外形与浏览窗口相似, 按行和列的形式显示数据。一个表格对象由若干列对象 (column) 组成, 每个列对象包含一个标头对象 (head) 和若干控件。这里的表格、列、标头和控件都有自己的属性与方法, 这使得用户对表格的控制变得更加灵活。

(1) 表格设计的基本操作。一旦指定了表格的数目, 就可有两种方法来调整表格的行高与列宽。一是通过设置表格的 HeaderHeight 和 RowHeight 属性调整行高, 通过设置列对象的 Width 属性调整列宽; 二是让表格处于编辑状态下, 然后通过鼠标拖动操作来可视地调整表格的行高与列宽。

在表格编辑状态下, 表格的周围有一个粗框, 要切换到表格的设计方式, 可选择表单或其他控件。

表格设计也可以调用表格生成器来进行。通过表格生成器能够交互地快速设置表格的有关属性, 创建所需要的表格。



操作步骤如下。

通过“表单控件”工具栏在表单上拖放一个表格。

用鼠标右键单击表格，在弹出的快捷菜单中选择“生成器”命令。

此时弹出“Grid Builder (表格生成器)”对话框，如图 6-6 所示，然后设置相关参数，完成后单击“OK (确定)”按钮返回。



图 6-6 “Grid Builder (表格生成器)”对话框

其中 4 个选项卡的作用如下。

- “Grid Items (表格项)”选项卡，指明要显示的字段。
- “Style (样式)”选项卡，指定表格的样式，如标准型、专业型和账务型等。
- “Layout (布局)”选项卡，指明各列的标题和控件类型，调整各列列宽。
- “Relationship (关系)”选项卡，设置一个一对多的关系，指明父表中的关键字与子表中的相关索引。

## (2) 表格属性。

常用的表格属性。

- RecordSourceType 属性与 RecordSource 属性。RecordSourceType 表明表格数据源的类型，RecordSource 属性指定表格数据源。
- ColumnCount 属性，指定表格的列数，即一个表格对象所包含的列对象的数目。
- LinkMaster 属性，用于指定表格控件中所显示的子表的父表名称。
- ChildOrder 属性，用于指定为建立一对多的关联关系，子表所要用的索引。
- RelationalExpr 属性，确定基于主表字段的关联表达式。

常用的列属性。

- ControlSource 属性，指定要在列中显示的数据源，通常为表的一个字段。
- CurrentControl 属性，指定列对象中的一个控件，该控件用于显示和接收列中活动单元格数据。列中非活动的单元格数据将在缺省的 TextBox 中显示。
- Sparse 属性，用于确定 CurrentControl 属性是影响列中的所有单元格还是只影响活动单元格。如果属性值是.T.，则只有列中的活动单元格使用由 CurrentControl 属性指定的控件显示和接收数据，其他单元格的数据用默认的 textbox 显示，如果属性值是.F.，那么列中所有的单元格都使用 CurrentControl 属性指定的控件显示数据，

活动单元格可接收数据。

常用的表头属性。

- Caption 属性，指定表头对象的标题文件，显示于列顶部。
- Alignment 属性，指定标题文本在对象中显示的对齐方式。其属性值与相应对齐方式如表 6-3 所示。

表 6-3 Alignment 属性值与相应对齐方式

| 属 性 值 | 对 齐 方 式 | 属 性 值 | 对 齐 方 式 |
|-------|---------|-------|---------|
| 0     | 居中靠左    | 1     | 居中靠右    |
| 2     | 居中      | 3     | 自动      |
| 4     | 左上对齐    | 5     | 右上对齐    |
| 6     | 中上对齐    | 7     | 左下对齐    |
| 8     | 右下对齐    | 9     | 中下对齐    |

## 10. 页框控件

页框控件是包含页面（page）的容器对象，而页面本身也是一种容器，其中可以包含所需要的控件。利用页框、页面和相应的控件可以构建选项卡对话框，这种对话框包含若干选项卡，选项卡就对应着这里所讲的页面。

页框定义了页面的总体特性，包括大小、位置、过界类型及哪页是活动的。页框中的页面相对于页框的左上角定位，并随页框在表单中的移动而移动。

在表单设计器环境下，往表单中添加页框的方法与添加其他控件的方法相同。在默认情况下，添加的页框包含两个页面，它们的标签为 page1 与 page2。用户可以通过设置页框的 pagecount 属性重新定义页面数目，通过定义页面的 caption 属性重新定义标签文本。

如果要向页面中添加控件，可以遵循以下步骤：首先，用鼠标右键单击页框，在弹出的快捷菜单中选择“编辑”命令，然后单击相应的页面标签，使该页面成为活动的；其次，在“表单控件”工具栏中选择需要的控件，并在页面中调整大小。

其主要属性如下。

（1）PageCount 属性，用于指明一个页框对象所包含的页（page）对象的数量。PageCount 的最大值为 99，最小值为 0。

（2）pages 属性是一个数组，用于存取页框中的某个对象。

（3）tabs 属性，指定页框中是否显示页面标签栏。

（4）abstretch 属性。如果页面的标签太长，标签栏无法在指定宽度的页框中显示出来，可以通过该属性指明其行为方式。如果设为 0，则为多重行；如果设为 1，则在一行内显示。

（5）ActivePage 属性，返回页框活动页的页号，或者使页框中的指定页成为活动的。

### 6.1.4 面向对象的概念

VFP 在支持 xBASE 传统的结构化程序设计的同时，扩展了面向对象程序设计（Object-Oriented-Programming）的新特点，后者具有更强大的功能和灵活性。

面向对象的设计方法和技术在人们关注的焦点方面不同于标准的过程化程序设计，现在编程人员不再单纯考虑代码从第一行到最后一行的编写，而考虑由建立对象提供给用户所有的功能。





## 1. 对象

所谓“对象 (Object)”，是将某一数据和使用该数据的一组基本操作或过程封装在一起，将此封装体看做一个实体。“面向对象”的基本思想就是把要构造的系统表示为对象的集合。

在结构化程序设计中，模块是功能的抽象，每个模块都是一个处理单位，它有输入和输出，而对象是包括数据、操作的整体，它将数据和功能抽象和统一。可以说，对象包括了模块。

在 VFP 中，表单 (forms)、表单集 (form sets) 和控件 (controls) 是包含在应用程序中的对象，可以通过对象的属性 (properties)、事件 (events) 和方法 (methods) 处理对象。VFP 面向对象这一语言的扩展可快速建立起应用系统，提供了对应用程序中对象的更多控制。这些扩展也使得建立和维护可重用代码更容易，可快速建立起应用系统，并具有如下优点。

- (1) 更紧凑的代码。
- (2) 不用精心策划，在应用程序中更易并入代码。
- (3) 降低了不同文件代码结合进应用程序的复杂度。

## 2. 类

面向对象的程序设计是通过对类、子类和对象等的设计来体现的，类是面向对象程序设计技术的核心，类就是定义了对象特征及对象外观和行为的模板。

类具有如下特征。

- (1) 封装性。隐藏了类中对象的内部数据或操作细节，只看到外表信息。
- (2) 继承性。任何类都可以从其他已有的类中派生，新类保持了父类的行为和属性，但增加了新的功能，体现了面向对象设计方法的共享机制。
- (3) 多态性。相同的操作可作用于多种类型的对象上，并获得不同的结果。
- (4) 抽象性。提取一个类或对象与众不同的特征，而不对该类或对象的所有信息进行处理。

## 3. 相关概念

(1) 基类。系统本身提供的类称为基类。基类包括容器类和控件类。容器类是其他对象的集合，如表格、单选按钮组；控件类是单一的对象，不包含其他对象，如命令按钮、文本框。

(2) 子类。以其他类定义为起点，对某一对象所建立的新类。新类将继承任何对父类所做的修改。

(3) 类库。存储类的文件，每个以可视方式设计的类都存储在一个类库中。类库文件的扩展名为 .vcx。

- (4) 属性。控件、字段或数据库对象的特性。
  - (5) 事件。由对象识别的一个动作。
  - (6) 方法。与对象相联系的由程序执行的一个处理过程，即处理事件的一组代码。
- 用户不可以为对象创建新的事件，但可以创建新的方法程序。

### 6.1.5 Visual FoxPro 基类

从 Visual FoxPro 基类中可以派生出子类或创建对象。Visual FoxPro 基类属性如表 6-4 所示。

Visual FoxPro 基类是系统本身内含的，并不存在于某个类库中。用户可以基于基类生成所需要的对象，也可以扩展基类创建自己的类。

表 6-4 Visual FoxPro 基类属性

| 属 性          | 说 明                                                                    |
|--------------|------------------------------------------------------------------------|
| Class        | 该类属于何种类型                                                               |
| BaseClass    | 该类由何种基类派生而来，例如 Form、CommandButton 或 Custom 等                           |
| ClassLibrary | 该类从属于哪种类库                                                              |
| ParentClass  | 对象所基于的类。若该类直接由 Visual FoxPro 基类派生而来，则 ParentClass 属性值与 BaseClass 属性值相同 |

### 6.1.6 Visual FoxPro 容器类与控件类

Visual FoxPro 中的类一般可分为两种类型：容器类与控件类。

容器类可以包含其他对象，并且允许访问这些对象。每种容器类所包含的对象如表 6-5 所示。

表 6-5 每种容器类所包含的对象

| 容 器   | 能包含的对象                            |
|-------|-----------------------------------|
| 命令按钮组 | 命令按钮                              |
| 容器    | 任意控件                              |
| 控件    | 任意控件                              |
| 表单集   | 表单、工具栏                            |
| 表单    | 页框、任意控件、容器或自定义对象                  |
| 表格列   | 表头对象及除表单、表单集、工具栏、计时器和其他列对象以外的任意对象 |
| 表格    | 表格列                               |
| 单选按钮组 | 单选按钮                              |
| 页框    | 页面                                |
| 页面    | 任意控件、容器和自定义对象                     |
| 工具栏   | 任意控件、页框和容器                        |

### 6.1.7 常用事件

事件是一种由系统预先定义而由用户或系统发出的动作。事件作用于对象，对象识别事件并做出相应的反应。事件可由系统或用户引发。Visual FoxPro 事件集如表 6-6 ~ 表 6-9 所示。

表 6-6 Visual FoxPro 基类的最小事件集

| 事 件     | 说 明                  |
|---------|----------------------|
| Init    | 当对象创建时激活             |
| Destroy | 当对象从内存中释放时激活         |
| Error   | 当类中的事件或方法在过程中发生错误时激活 |



表 6-7 Visual FoxPro 中的常用事件分类

| 事 件 类 型   | 事 件 名 称                                               |
|-----------|-------------------------------------------------------|
| 鼠标事件      | Click DblClick RightClick DropDown DownClick UpClick  |
| 键盘事件      | KeyPress                                              |
| 改变控件内容的事件 | InteractiveChange                                     |
| 控件焦点的事件   | GotFocus LostFocus When Valid                         |
| 表单事件      | Load Unload Destroy Activate Resize Paint QueryUnload |
| 数据环境事件    | After CloseTable Before OpenTable                     |
| 项目事件      | QueryModifyFile 等                                     |
| OLE 事件    | OLECompleteDrag 等                                     |
| 其他事件      | Timer Init Destroy Error                              |

表 6-8 一些核心事件和触发时间

| 事 件                | 触 发 时 间          |
|--------------------|------------------|
| Load               | 当表单或表单集被加载时产生    |
| Unload             | 当表单或表单集从内存中释放时产生 |
| Init               | 创建对象时产生          |
| Destroy            | 从内存中释放对象时产生      |
| Click              | 用户在对象上单击鼠标时产生    |
| DblClick           | 用户在对象上双击鼠标时产生    |
| RightClick         | 用户在对象上单击鼠标右键时产生  |
| GetFocus           | 对象得到焦点时产生        |
| LostFocus          | 对象失去焦点时产生        |
| KeyPress           | 用户按键时产生          |
| MouseDown          | 在对象上按下鼠标时产生      |
| MouseUp            | 在对象上释放鼠标指针时产生    |
| MouseMove          | 在对象上移动鼠标指针时产生    |
| InteractiveChange  | 交互式改变对象值时产生      |
| ProgrammaticChange | 可编程地改变对象值时产生     |

表 6-9 Visual FoxPro 中的常用方法

| 名 称       | 调 用 语 法                               | 功 能                                            |
|-----------|---------------------------------------|------------------------------------------------|
| AddObject | Object.AddObject (cName,cClass[,...]) | 在运行时向容器对象中添加对象                                 |
| Clear     | Object.Clear                          | 清除组合框或列表框控件中的内容                                |
| Hide      | Object.Hide                           | 通过把 Visible 属性设置为 .F., 来隐藏表单、表单集或工具栏           |
| Show      | Object.Show                           | 把 Visible 属性设置为 .T., 显示并激活一个表单或表单集, 并确定表单的显示模式 |
| Refresh   | Object.Refresh                        | 重画表单或控件, 并刷新所有值                                |
| Release   | Object.Release                        | 从内存中释放表单或表单集                                   |
| Quit      | Object.Quit                           | 结束一个 Visual FoxPro 9.0 实例, 返回到创建它的应用程序         |

## 任务 6.2 实例训练

在建立表单的过程中, 要定义控件的属性, 确定事件或方法的代码。在表单设计中, 两个最基本的可视对象为容器和控件。表单就是一个容器, 它可以容纳多个控件。

### 6.2.1 利用表单向导创建表单

操作步骤如下。

- (1) 在 Visual FoxPro 主窗口下, 选择“文件”|“新建”命令, 在弹出的“新建”对话框中, 选中“表单”单选按钮, 如图 6-7 所示。
- (2) 单击“向导”按钮, 进入“Wizard Selection (向导选取)”对话框, 如图 6-8 所示。



图 6-7 “新建”对话框

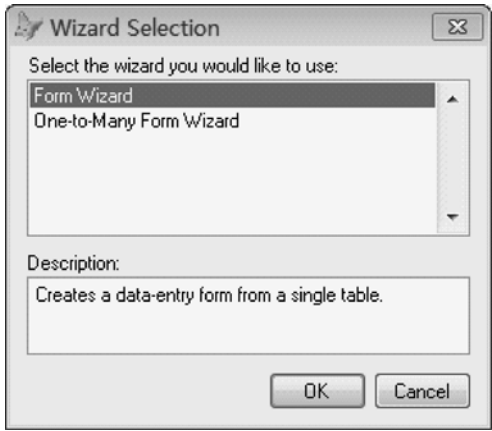


图 6-8 “Wizard Selection (向导选取)”对话框

其中“Form Wizard(表单向导)”是为单个表创建操作数据的表单,“One-to-Many Form Wizard(一对多表单向导)”适合创建基于两个具有一对多关系的表的表单。

- (3) 单击“OK(确定)”按钮, 进入表单向导第一步操作, 选择相关的表与字段, 如图 6-9 所示。

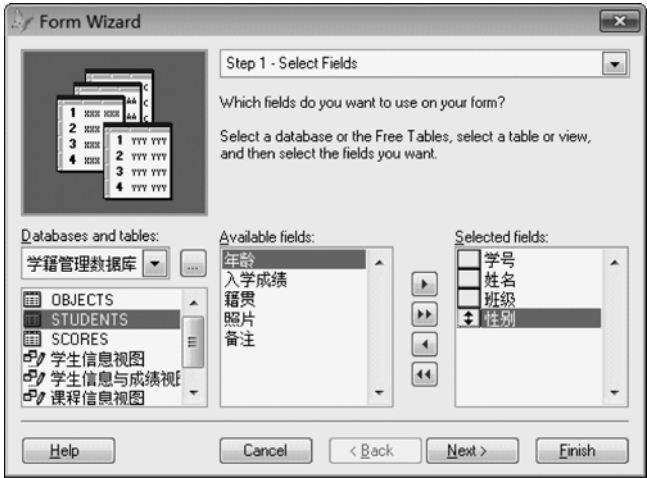


图 6-9 表单向导第一步



(4) 选择好相应的表与字段后,单击“Next(下一步)”按钮,弹出如图 6-10 所示的表单向导第二步对话框。

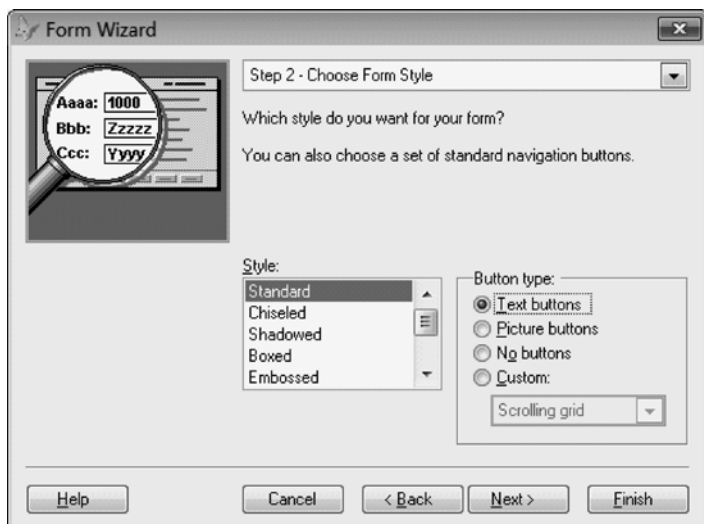


图 6-10 表单向导第二步

(5) 选择好相应的表单样式之后,单击“Next(下一步)”按钮,弹出如图 6-11 所示的表单向导第三步对话框。

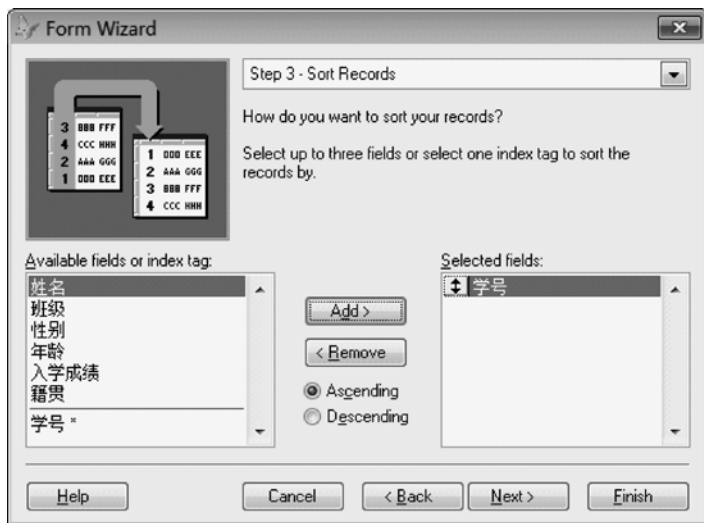


图 6-11 表单向导第三步

在这一步中,我们选择排序字段与排序方式。例如,添加“学号”字段作为排序字段。如果是多字段,可继续添加,但最多不超过 3 个。

(6) 单击“Next(下一步)”按钮,来完成表单向导。在此为表单命名并选中“Save and run form(保存并运行表单)”单选按钮,单击“Finish(完成)”按钮,完成表单,表单向导第四步如图 6-12 所示。

(7) 为表单文件命名,假设名称为 students.scx,单击“保存”按钮保存表单文件,如图 6-13 所示。



图 6-12 表单向导第四步



图 6-13 保存表单文件

Visual FoxPro 系统将自动生成表单，如图 6-14 所示。可以通过表单下面的按钮来查看 students 表中的相关数据。



图 6-14 生成表单



创建基于多表的表单与此类似，这里不再举例说明。

## 6.2.2 利用表单设计器创建表单

利用表单设计器是创建表单的另一种方法。

可以使用下列 3 种方法中的任意一种来调用表单设计器。

### 1. 在项目管理器下调用

操作步骤如下。

(1) 在“项目管理器”窗口中选择“文档”选项卡，然后选择其中的“表单”，“项目管理器”窗口如图 6-15 所示。



图 6-15 “项目管理器”窗口

(2) 单击“新建”按钮，系统弹出“新建表单”对话框，如图 6-16 所示。

(3) 单击“新建表单”按钮，进入表单设计器界面，如图 6-17 所示。



图 6-16 “新建表单”对话框



图 6-17 表单设计器界面

## 2. 通过菜单调用

操作步骤如下。

- (1) 选择“文件”|“新建”命令，打开“新建”对话框。
- (2) 选择“表单”文件类型，然后单击“新建”按钮。

## 3. 通过命令调用

在命令窗口中输入 `CREATE FORM [ <文件名> ]` 命令。

### 6.2.3 表单的修改与运行

#### 1. 修改表单

一个表单无论是通过何种途径创建的，都可以使用表单设计器加以编辑。可采用以下方法打开表单文件来进入表单设计器界面。

- (1) 选择“文件”|“打开”命令，打开相关表单文件。
- (2) 在命令窗口中输入命令 `MODIFY FORM <表单文件名>`。

#### 2. 运行表单

在 Visual FoxPro 系统中，以下几种方式可以运行表单。

- (1) 在项目管理器中，选择要运行的表单，然后单击“Run (运行)”按钮。
- (2) 在表单设计界面下，选择“表单”|“执行表单”命令，或者在对话框中指定要运行的表单文件并单击“运行”按钮。
- (3) 选择“程序”|“运行”命令，打开“运行”对话框，然后在对话框中指定要运行的表单文件并单击“运行”按钮。

- (4) 在命令窗口中输入以下命令。

```
DO FORM <表单文件名>[NAME<变量名>]
WITH <实参 1>[,<实参 2>,...][LINKED][NOSHOW]
```

使用说明如下。

包含 NAME 子句，在表单运行引发 Init 事件时，系统会将各实参的值传递给该事件代码 PARAMAETERS 或 LPARAMTERS 子句中的各个形参。

包含 LINKED 关键字，表单对象将随着指向它的变量的清除而关闭。

运行表单时，一般在产生表单对象后，调用表单对象的 SHOW 方法将显示于表单。如果包含“NOSHOW”关键字，则表单运行时将不显示，直至表单对象的 Visible 属性设置为.T.，或者是调用了 Show 方法。

表单运行后，可单击标准工具栏上的“修改表单”按钮，切换到表单设计器界面。

### 6.2.4 为表单建立数据环境

为表单建立数据环境，数据环境中能够包含与表单有联系的表和视图，以及表之间的关系。通常情况下，数据环境中的表或视图会随着表单的打开或运行而打开，并随着表单





的关闭或释放而关闭。

### 1. 常用的属性

AUTO OPEN TABLES: 当运行或打开表单时, 是否打开数据环境中的表和视图。

AUTO CLOSE TABLES: 当释放或关闭表单时, 是否关闭由数据库环境指定的表和视图。

它们的默认值为.T.

### 2. 打开数据环境设计器

在表单设计器界面下, 单击“表单设计器”工具栏上的“数据环境”按钮, 即可打开“数据环境设计器”窗口, 如图 6-18 所示。



图 6-18 “数据环境设计器”窗口

### 3. 向数据环境添加表或视图

(1) 选择“数据环境” “添加”命令, 或者右击“数据环境设计器”窗口, 在弹出的快捷菜单中选择“添加”命令, 打开“添加表或视图”对话框, 如图 6-19 所示。

(2) 选择要添加的表或视图并单击“添加”按钮。

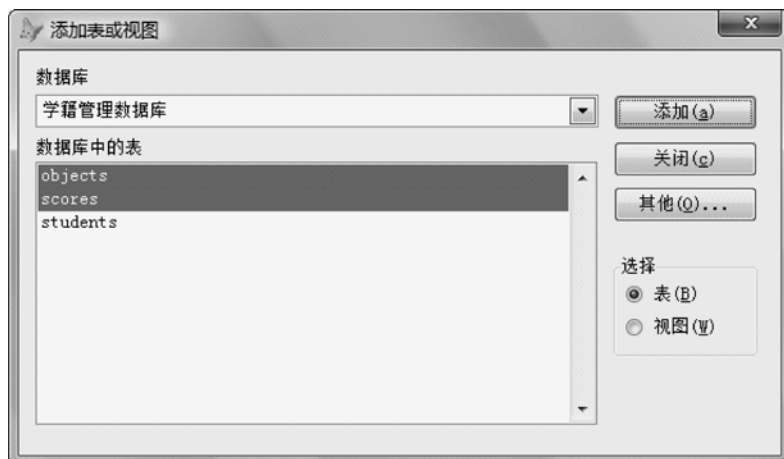


图 6-19 “添加表或视图”对话框

注意：如果是第一次打开数据环境设计器，即数据环境设计器中没有表，则在打开时，“数据环境设计器”和“添加表或视图”窗口会同时弹出，需要先进行添加表操作。

#### 4. 从数据环境移去表或视图

(1) 在“数据环境设计器”窗口中，选择要移去的表或视图。

(2) 选择“数据环境” “移除”命令。

也可以用鼠标右键单击要移去的表或视图，在弹出的快捷菜单中选择“移除”命令。当表从数据环境中移去时，与这个表有关的所有关系也将随之消失。

#### 5. 向表单添加数据

Visual FoxPro 允许用户从“数据环境设计器”窗口、“项目管理器”窗口或“数据库设计器”窗口中直接将字段、表或视图拖入表单，系统将产生相应的控件字段与之对应。

例如，把 students 表中的“学号”与“姓名”字段拖入窗体中，如图 6-20 所示。



图 6-20 将字段拖入窗体


#### 6.2.5 给表单添加标签控件

【例 6-1】给表单添加 3 个标签控件 Lable1、Lable2 和 Lable3，用于显示系统程序设计欢迎界面。

操作步骤如下。

- (1) 选择“文件” “新建”命令，打开“新建”对话框。
- (2) 选择“表单”文件类型，然后单击“新建”按钮。
- (3) 进入“表单设计器”界面。



(4) 在“表单控件”工具栏中,单击标签按钮 , 在表单的合适位置上拖放或单击鼠标左键, 将 3 个标签控件加入到表单中, 如图 6-21 所示。

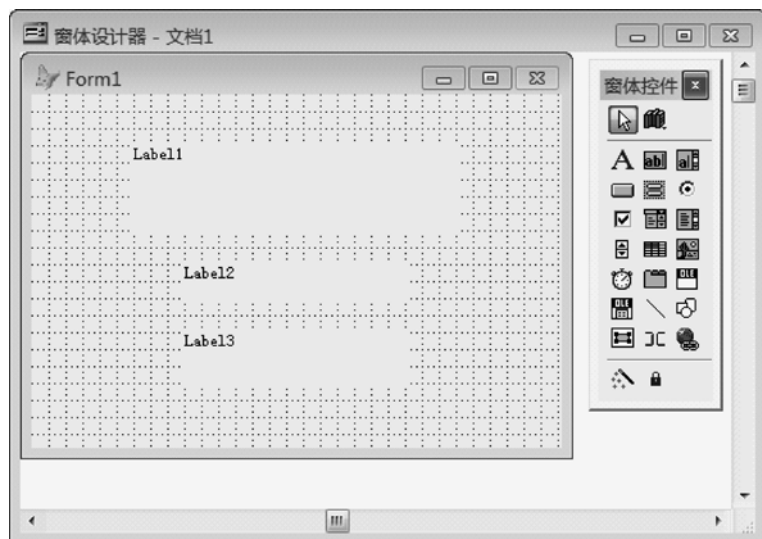


图 6-21 将标签控件加入到表单中

(5) 在 Visual FoxPro 系统的主菜单下,单击“属性窗口”按钮,打开属性窗口。

对 Label1 属性进行定义。

Caption——欢迎使用学生学籍管理系统！

Top——36

Left——30

Height——29

Width——314

BackColor——192,192,192

Backstyle——0-Transparent

Autosize——.T.-True

Fontname——隶书

FontSize——18

Name——Label1

对 Label2 属性进行定义。

Caption——版权所有 (C) 2011

Top——96

Left——132

Height——16

Width——104

BackColor——192,192,192

Backstyle——1-Opaque (Default)

Autosize——.T.-True

Fontname——宋体

Fontsize——10

Fontcolor——255,0,0

Name——Label2

对 Lable3 属性进行定义。

Caption——开发人：\*\*\*

Top——132

Left——155

Height——17

Width——75

BackColor——192,192,192

Backstyle——1- Opaque ( Default )

Autosize——.T.-True

Fontname——宋体

Fontsize——10

Fontcolor——0,0,255

Name——Label3

3 个标签定义完成以后，保存表单 form1.scx，得到如图 6-22 所示的界面。

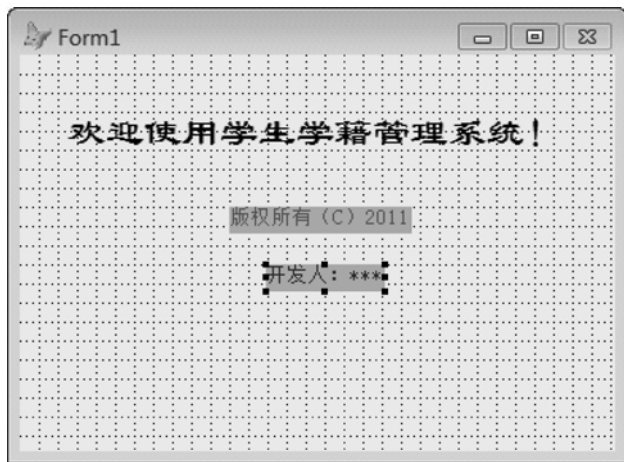


图 6-22 保存表单 form1.scx 后的界面

### 6.2.6 给表单添加命令按钮控件

【例 6-2】设计一个包含命令按钮控件的表单 Form2，在 Form1 的基础上，添加两个具有不同功能的命令按钮 Command1、Command2 控件，用来控制系统登录表单的操作。

操作步骤如下。

- (1) 打开 Form1，进入表单设计器窗口。
- (2) 在属性窗口下定义 Form1 的属性 Caption 为 Form2。
- (3) 在“表单控件”工具栏中重复选择命令按钮控件，拖动鼠标到表单的合适位置上，将两个命令按钮控件加入到表单中。
- (4) 定义两个命令按钮控件 Command1、Command2 的相关属性。在属性窗口中定义



Command1 的主要属性 Caption 为“进入系统”；在属性窗口中定义 Command2 的主要属性 Caption 为“退出系统”，得到如图 6-23 所示的定义属性后的界面。

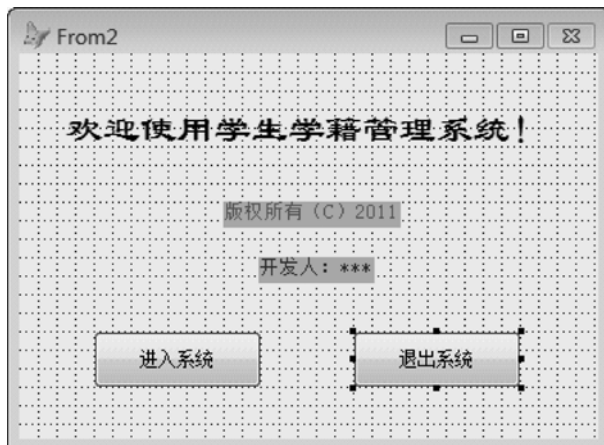


图 6-23 定义属性后的界面

(5) 在 Visual FoxPro 系统的主菜单下，选择“显示”“代码”命令，或者双击“命令按钮”控件，进入代码编辑窗口，如图 6-24 所示，在代码编辑窗口中输入代码。



图 6-24 代码编辑窗口

Command1 的 Click 事件代码如下。

```
Do form f:\学生学籍管理系统\students.scx 或
Do form students.scx
```

Command2 的 Click 事件代码如下。

```
A= messagebox("要退出学生学籍管理系统?",4+16+0, "对话框")
If a=6
 Release thisform
Endif
```

(6) 保存表单 form2.scx 并运行，界面如图 6-25 所示。

如果单击“进入系统”按钮，则进入表单 students.scx，如图 6-26 所示。

如果单击“退出系统”按钮，则弹出如图 6-27 所示的对话框，单击“是”按钮则退出系统。



图 6-25 保存表单 form2.scx 并运行后的界面



图 6-26 单击“进入系统”按钮后的界面



图 6-27 对话框

### 6.2.7 给表单添加文本框控件

**【例 6-3】** 设计一个包含文本框控件的表单 Form3，用于系统口令的输入。  
操作步骤如下。

- (1) 选择“文件” “新建”命令，打开“新建”对话框。
- (2) 选择“表单”文件类型，然后单击“新建”按钮。
- (3) 进入“表单设计器”界面。
- (4) 调出“属性”窗口，定义表单属性。

Caption——系统登录

BackColor——236,233,216

- (5) 在“表单控件”工具栏中，单击标签控件按钮，添加一个标签控件，定义其主要属性。



Caption——口令

FontSize——12

(6) 在“表单控件”工具栏中,单击两次命令按钮控件,添加两个命令按钮控件 Command1 与 Command2。

定义 Command1 的主要属性。

Caption——确认

定义 Command2 的主要属性。

Caption——退出

(7) 在“表单控件”工具栏中,单击“文本框”控件,在表单中添加一个文本框控件,定义其主要属性。

Passwordchar——\*

Value——无

Name——upassword

(8) 以上控件定义完成后,各控件的具体布局如图 6-28 所示。

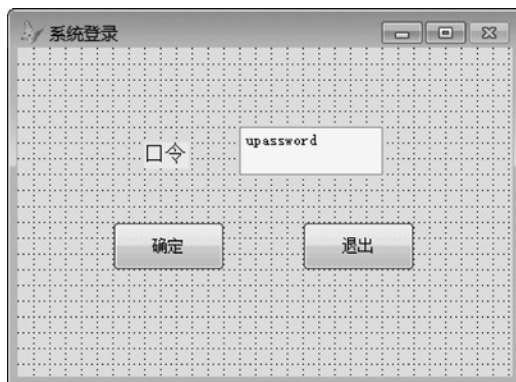


图 6-28 表单设计器中的控件布局

(9) 双击“确定”按钮来确定其 Click 事件代码。

```
If TRIM(thisform.upassword.value) <> "root"
 If messagebox("口令错误,请重新输入口令",0+16,"提示")=1
 Thisform.upassword.setfocus
 ENDIF
Endif
If TRIM(thisform.upassword.value) = "root"
 DO FORM f:\学生学籍管理系统\students.scx
Endif
```

(10) 双击“退出”按钮来确定其 Click 事件代码。

```
A=MESSAGEBOX("要退出系统登录?",4+16+0,"对话框口")
If A=6
 RELEASE THISFORM
Endif
```

(11) 运行表单,登录窗口如图 6-29 所示。



图 6-29 “系统登录”窗口

### 6.2.8 给表单添加编辑框控件

【例 6-4】设计一个表单 form4.scx，包含一个编辑框控件和两个命令按钮控件。

操作步骤如下。

(1) 选择“文件” “新建”命令，打开“新建”对话框。

(2) 选择“表单”文件类型，然后单击“新建”按钮。

(3) 进入“表单设计器”界面。

(4) 设置编辑框控件主要属性。

HideSelection 为.F，这样字符串就会显示为选定的状态。

(5) 加入两个命令按钮 Command1 与 Command2，定义其属性。

对于命令按钮 Command1，Caption 为“查找”；对于命令按钮 Command2，Caption 为“替换”。

得到的表单如图 6-30 所示。

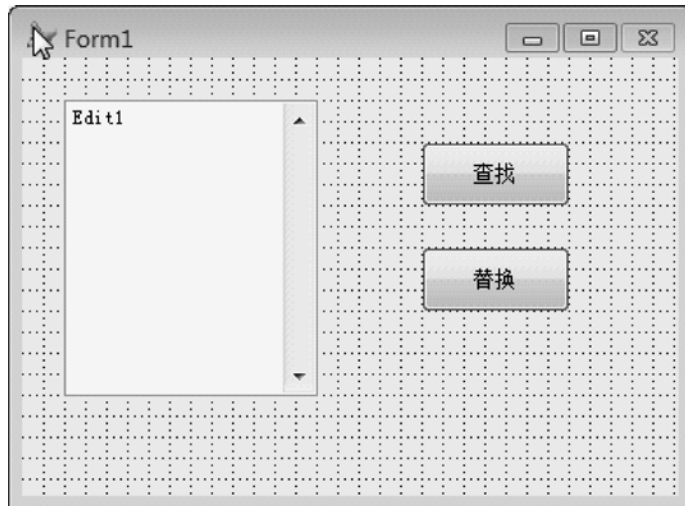


图 6-30 表单效果





(6) 双击按钮来确定其 Click 事件。  
“查找”按钮的 Click 事件代码如下。

```
N=at("book",thisform.edit1.value)
If n<>0
Thisform.edit1.selstart=n-1
Thisform.edit1.sellength=len("book")
Else
Wait windows "没有相匹配的词" timeout 1
Endif
```

“替换”按钮的 Click 事件代码如下。

```
If thisform.edit1.seltext="book"
Thisform.edit1.seltext="pen"
Else
Wait windows "没有选择要替换的词" timeout 1
Endif
```

(7) 表单执行结果——查找,如图 6-31 所示;表单执行结果——替换,如图 6-32 所示。



图 6-31 表单执行结果——查找

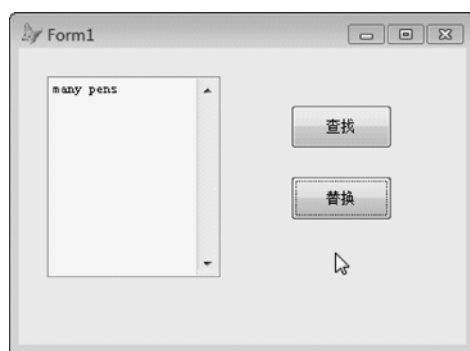


图 6-32 表单执行结果——替换

## 6.2.9 设计查询统计表单

【例 6-5】依据前面建立的学生表 students.dbf、成绩表 scores.dbf,设计一个能进行查询统计的表单 cxtj.scx,如图 6-33 所示。当输入姓名并单击“统计”按钮时,会在右边的表格内显示该同学所选课程的成绩,并在左边的文本框内显示该学生的最高分、最低分和平分。单击“退出”按钮将关闭表单。

操作步骤如下。

- (1) 新建表单 cxtj.scx,在表单上添加如图 6-33 所示的控件,并设置其 Caption 属性。
- (2) 将表格的 ColumnCount(列的数目)属性设置为 2,RecordSourceType 属性值设置为“4-SQL Statement 说明”,并将其两个列标头设置为“课程号”、“成绩”。
- (3) 设置“统计”按钮的 Click 事件代码。(请读者自己完成。)
- (4) 设置“退出”按钮的 Click 事件代码。(请读者自己完成。)
- (5) 保存表单,关闭表单设计器窗口。

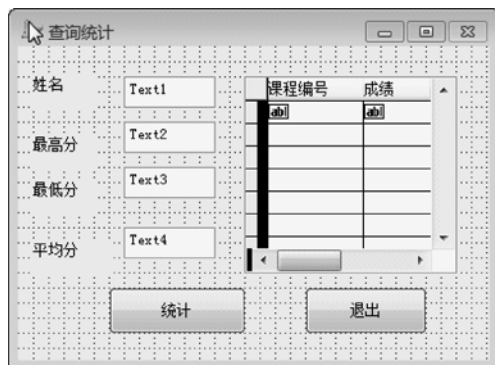


图 6-33 “查询统计”表单

## 任务 6.3 项目分析

这个项目中，我们建立了学生学籍管理系统需要的一些表单，包括“欢迎进入系统”表单、“系统登录”表单、“学生信息”表单和“查询统计”表单等。下面我们依次来查看它们的界面。

### 6.3.1 “欢迎进入系统”表单

“欢迎进入系统”表单，如图 6-34 所示。



图 6-34 “欢迎进入系统”表单

### 6.3.2 “系统登录”表单

“系统登录”表单，如图 6-35 所示。



图 6-35 “系统登录”表单



### 6.3.3 “学生信息”表单

“学生信息”表单，如图 6-36 所示。

图 6-36 “学生信息”表单

### 6.3.4 “查询统计”表单

“查询统计”表单，如图 6-37 所示。

图 6-37 “查询统计”表单

## 任务 6.4 模仿设计与自主设计

### 6.4.1 “欢迎使用图书管理系统”表单

为图书管理系统设计一个“欢迎使用图书管理系统”表单，要求如下。

- (1) 包括欢迎语句“欢迎使用图书管理系统”。
- (2) 包括制作人的信息及制作时间。
- (3) 包括“进入”和“退出”按钮。

#### 6.4.2 “图书信息”表单

为图书管理系统设计一个“图书信息”表单，用于显示所有图书的信息，要求如下。

- (1) 界面美观，可插入背景图。
- (2) 可以查看每条记录，并可以进行查找、添加和删除等操作。

#### 6.4.3 “借阅者信息查询统计”表单

为图书管理系统设计一个“借阅者信息查询统计”表单，用于查询统计借阅者的信息，要求如下。

- (1) 可以查询读者信息。
- (2) 可以统计尚未还书的读者的信息。

### 项目小结 6

项目 6 介绍了 Visual FoxPro 面向对象的若干基本概念、表单的创建和管理、表单设计器环境及在该环境下的一些操作，并详细介绍了表单设计器常用控件的使用方法。这个项目也是本教材的重点章节，读者应着重掌握。

### 项目习题 6

#### 1. 选择题

- (1) 在 Visual FoxPro 中，运行表单 T1.scx 的命令是 ( )。
  - A . DO T1
  - B . RUN FORM1 T1
  - C . DO FORM T1
  - D . DO FORM T1
- (2) 在 Visual FoxPro 中，为了将按钮的 Click 事件代码设置为表单从内存中释放（清除），可为该按钮的 Click 事件添加命令 ( )。
  - A . ThisForm.Refresh
  - B . ThisForm.Delete
  - C . ThisForm.Hide
  - D . ThisForm.Release
- (3) 假定一个表单里有一个文本框 Text1 和一个命令按钮组 CommandGroup1，命令按钮组是一个容器对象，其中包含 Command1 和 Command2 两个命令按钮。如果要在 Command1 命令按钮的某个方法中访问文本框的 Value 属性值，下面哪个式子是正确的？ ( )
  - A . ThisForm.Text1.value
  - B . This.Parent.value
  - C . Parent.Text1.value
  - D . This.Parent.Text1.value
- (4) 下面是关于表单数据环境和数据环境中两个表的关系的叙述，其中正确的是 ( )。
  - A . 数据环境是对象，关系不是对象
  - B . 数据环境不是对象，关系是对象



C. 数据环境是对象, 关系是数据环境中的对象

D. 数据环境和关系都不是对象

(5) 下面关于表单若干常用事件的描述中, 正确的是 ( )。

A. 释放表单时, Unload 事件在 Destroy 事件之前引发

B. 运行表单时, Init 事件在 Load 事件之前引发

C. 单击表单的标题栏, 引发表单的 Click 事件

D. 上面的说法都不对

(6) 假设表单上有一个选项组: 男 女, 其中第一个单选按钮“男”被选中。请问该选项组的 Value 属性值为 ( )。

A. .T.

B. "男"

C. 1

D. "男"或 1

(7) 在设计代码时, 应该用 ( ) 属性值而不能用 ( ) 属性值来引用对象。

A. Name、Caption

B. Alignment、Name

C. Caption、Name

D. Alignment、Caption

(8) 在 Visual FoxPro 中, Default 属性值为.T.的命令按钮为 ( ) 按钮。

A. 确定

B. 否定

C. 确认

D. 否认

(9) ( ) 属性用来指定表单或控件能否响应由用户引发的事件。

A. Enabled

B. Visible

C. Cancel

D. Default

(10) 如果想使一个选项组中包括 3 个单选按钮, 可将 ( ) 属性值设置为 3。

A. Value

B. ButtonCount

C. ControlSource

D. Buttons

## 2. 填空题

(1) 在表单中确定控件是否可见的属性是\_\_\_\_\_。

(2) 在 Visual FoxPro 的表单设计中, 为表格控件指定数据源的属性是\_\_\_\_\_。

(3) 在 Visual FoxPro 中为表单指定标题的属性是\_\_\_\_\_。

(4) 在 Visual FoxPro 中表单的 Load 事件发生在 Init 事件之\_\_\_\_\_。

(5) 在面向对象方法中, \_\_\_\_\_描述的是具有相似属性与操作的一组对象。



# 项目 7

## 设计学生学籍管理 系统的菜单

### 项目要点◎

---

- 下拉式菜单设计。
- 快捷菜单设计。
- 指定菜单任务。
- 运行菜单。

### 学习目标◎

---

- 理解并掌握菜单的功能及结构。
- 理解并掌握菜单的建立方法与应用。

在一个好的系统程序中，菜单起着组织协调其他对象的关键作用，一个好的菜单系统会给用户一个友好的界面，并带来操作上的便利，对数据库的操作而言，更是如此。本项目将首先介绍 Visual FoxPro 系统菜单的基本情况，然后介绍如何定制系统菜单，如何设计下拉式菜单和快捷菜单。

### 任务 7.1 熟悉菜单基础知识

了解 Visual FoxPro 系统菜单的结构、特点和行为，是设计用户菜单系统的基础。



### 7.1.1 菜单系统的结构

Visual FoxPro 支持两种类型菜单：条形菜单和弹出式菜单。每个条形菜单都有一个内部名称和一组菜单选项，每个菜单选项都有一个名称和内部名称。每个弹出式菜单也有一个内部名称和一组菜单选项，每个菜单选项都有一个名称和选项序号。菜单项的名称显示于屏幕供用户识别，菜单及菜单项的内部名称或选项序号则用于代码中的引用。

每个菜单选项都可以有选择地设置一个热键和一个快捷键。热键通常是一个字符，当菜单激活时，可以按菜单项的热键快速选择该菜单项。快捷键通常是由 Ctrl 键和另一个字符键组成的组合键。不管菜单是否激活，都可以通过快捷键选择相应的菜单选项。

无论是哪种类型的菜单，当选择其中某个选项时都会有一定的动作。这个动作可以是下面 3 种情况中的一种：执行一条命令，执行一个过程，激活一个菜单。

典型的菜单系统一般是一个下拉式菜单，由一个条形菜单和一组弹出式菜单组成。其中，条形菜单为主菜单，弹出式菜单为子菜单。当选择一个条形菜单选项时，激活相应的弹出式菜单。

快捷菜单一般由一个或一组上下级的弹出式菜单组成。

### 7.1.2 下拉式菜单设计过程

菜单系统是由菜单栏、菜单标题、菜单和菜单项组成的。

其中，菜单栏用于放置多个菜单标题；菜单标题是每个菜单的名称，单击某菜单标题可打开相应的菜单；菜单包含命令、过程、填充名称和子菜单；菜单项用来实现某一具体任务。

下拉式菜单是一种最常见的菜单，用 Visual FoxPro 提供的菜单设计器可以方便地进行下拉式菜单设计。具体说来，菜单设计器的功能有两个：一是为顶层表单设计下拉式菜单；二是通过定制 Visual FoxPro 系统菜单建立应用程序的下拉式菜单。

在利用菜单设计器设计菜单时，各菜单项及其功能既可以由自己来定义，也可以采用 Visual FoxPro 系统的标准菜单项及其功能。

菜单设计的过程如图 7-1 所示。

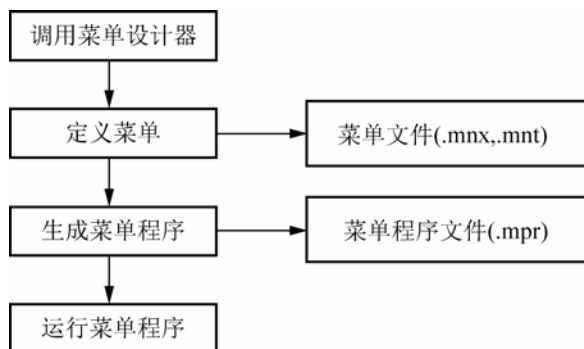


图 7-1 菜单设计的过程示意图

#### (1) 调用菜单设计器。

操作步骤如下。

选择“文件”|“新建”命令。

在“新建”对话框中选中“菜单”单选按钮，然后单击“新建”按钮，如图 7-2 所示。

在“新建菜单”对话框中单击“菜单”按钮，如图 7-3 所示。

这样就会弹出“菜单设计器”窗口，如图 7-4 所示。

(2) 定义菜单。在“菜单设计器”窗口中定义菜单，指定菜单的各项内容，如菜单项的名称、快捷键等。

指定完菜单的各项内容后，应将菜单定义保存到.mnx 文件中。



图 7-2 “新建”对话框



图 7-3 “新建菜单”对话框

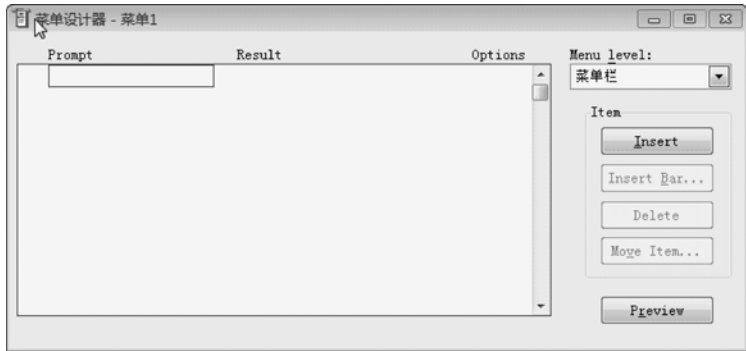


图 7-4 “菜单设计器”窗口

操作步骤：选择“文件”|“保存”命令或按“Ctrl+W”组合键。

(3) 生成菜单程序。定义菜单文件存放着菜单的各项定义，但其本身是一个表文件，并不能运行。生成菜单程序就是要根据菜单定义来产生可执行的菜单程序文件.mpr 文件。

操作步骤如下。

在菜单设计器环境下，选择“菜单”|“生成”命令。

在“生成菜单”对话框中指定菜单程序文件的名称与存放路径，最后单击“生成”按钮。





(4) 运行菜单程序。可使用命令 `do<文件名>` 运行菜单程序, 注意文件名扩展名 `.mpr` 不能省略。

### 7.1.3 下拉式菜单定义方法

#### 1. “菜单设计器”窗口

下面介绍如何在“菜单设计器”窗口中定义菜单。“菜单设计器”窗口每页显示和定义一个菜单, 可以是条形菜单, 也可以是弹出式菜单。“菜单设计器”窗口打开时, 首先显示和定义的是条形菜单。窗口的左侧是一个列表框, 其中每行定义当前菜单的一个菜单项, 包括“Prompt (提示)”、“Result (结果)”和“Option (选项)”3列内容。

##### (1) “Prompt (提示)”列。

定义菜单的名称, 也称为标题, 用于显示, 并非内部名称。在定义菜单名称时, 可以设置菜单项的访问键, 方法是在要作为访问键的字符前加上“<”两个字符。

例如, (<E), 那么字母 E 就为该菜单的访问键。

可以根据各菜单项功能来将弹出式菜单的菜单项分组, 如将剪切、复制和粘贴分为一组, 将查找、替换分为一组。系统提供的分组手段为在两组之间插入一条水平分组线, 方法是在相应的“提示”列上输入“-”两个字符。

##### (2) “Result (结果)”列。

该列用于指定当用户选择菜单项时的动作。该列主要有“命令”、“过程”、“子菜单”、“填充名称”或“菜单项”等。

命令: 选择“命令”, 列表框的右侧会出现一个文本框, 可以在文本框中输入一条具体命令, 当选择该菜单时, 则执行该命令。

过程: 选择此项时, 列表框的右侧会出现“创建”命令按钮, 单击“创建”按钮将打开一个文本编辑窗口, 可以在其中输入和编辑过程代码。

子菜单: 选择此项, 列表框的右侧就会出现“创建”或“编辑”按钮 (第一次显示为“创建”, 以后显示为“编辑”), 单击此按钮, “菜单设计器”窗口就会切换到子菜单页, 可以在其中定义子菜单。

##### (3) “Option (选项)”列。

每个菜单的“选项”列都有一个无符号按钮, 单击该按钮就会出现“提示选项”对话框, 如图 7-5 所示。

该对话框供用户定义菜单的其他属性。在对话框中定义过属性后, 按钮上就会出现“ ”号。

该对话框的主要属性如下。

快捷方式: 指定菜单的快捷键。

操作步骤: 先用鼠标单击“键标签”文本框, 使光标定于该文本框, 然后在键盘上按快捷键, 快捷键通常是 Ctrl 或 Alt 键与另一个字符键的组合。

要取消定义过的快捷键, 可以先用鼠标单击“键标签”文本框, 然后按空格键。

跳过: 定义菜单项的跳过条件。

指定一个表达式, 由表达式的值来决定该菜单项是否可选。当菜单激活时, 如果表达式的值是.T., 则菜单项以灰色显示, 表示不可用。



图 7-5 “提示选项”对话框

信息：定义菜单项的说明信息。

当鼠标指向该菜单项时，说明信息就显示在 Visual FoxPro 系统主窗口的状态栏上。

主菜单名或菜单项：指定条形菜单项的内部名称或弹出式菜单项的序号。

只有当菜单项的“结果”选为“命令”、“过程”或“子菜单”时，该文本框才有效。

“菜单设计器”窗口的其他按钮。

“Insert（插入）”按钮。单击该按钮，可在当前菜单项行之前插入一个新菜单项行。

“Insert Bar（插入栏）”按钮。在当前菜单项行之前插入一个 Visual FoxPro 系统菜单命令。

“Delete（删除）”按钮。删除当前菜单行。

“Move Item（移动项）”按钮。单击此按钮，可以改变菜单项在当前菜单中的位置。

“Preview（预览）”按钮。预览菜单效果。

## 2. “显示”菜单

在菜单设计器环境下，系统的“显示”菜单会出现两条命令：“常规选项”与“菜单选项”。

（1）“常规选项”对话框。选择“显示”|“常规选项”命令，就会打开“常规选项”对话框，如图 7-6 所示。

在该对话框中，可以定义整个下拉式菜单系统的总体属性。

编辑：为条形菜单指定相应的过程代码。

位置：指明正在定义的一个下拉式菜单与当前系统菜单的关系。

- 替换：用定义的菜单内容去替换当前系统菜单原有内容。
- 追加：将定义的菜单内容添加到当前系统菜单原有内容的后面。



图 7-6 “常规选项”对话框

- 在...之前：将定义的菜单内容添加到当前系统菜单某个弹出式菜单之前。
- 在...之后：将定义的菜单内容插入在当前系统菜单某个弹出式菜单之后。

菜单代码：不论选“设置”还是“清理”，都会打开一个相应的代码编辑窗口，“设置”代码在菜单产生之前执行，“清理”代码在菜单产生之后执行。

(2)“菜单选项”对话框。该对话框主要用于定义当前弹出式菜单的公共过程代码。

#### 7.1.4 快捷菜单设计

一般来讲，下拉式菜单作为一个应用程序的菜单系统，列出了整个程序所具有的功能，而快捷菜单一般从属于某个界面对象，当用鼠标右键单击该对象时，就会在单击处弹出快捷菜单。快捷菜单通常列出与处理的相应对象有关的一些功能命令。利用系统的快捷菜单设计器可以方便地定义与设计快捷菜单。与下拉式菜单相比，快捷菜单没有条形菜单，只有弹出式菜单。

操作步骤如下。

- (1) 选择“文件”|“新建”命令。
- (2) 在“新建”对话框中选中“菜单”单选按钮，然后单击“新建”按钮。
- (3) 在“新建菜单”对话框中单击“快捷菜单”按钮，进入“快捷菜单设计器”窗口，如图 7-7 所示。

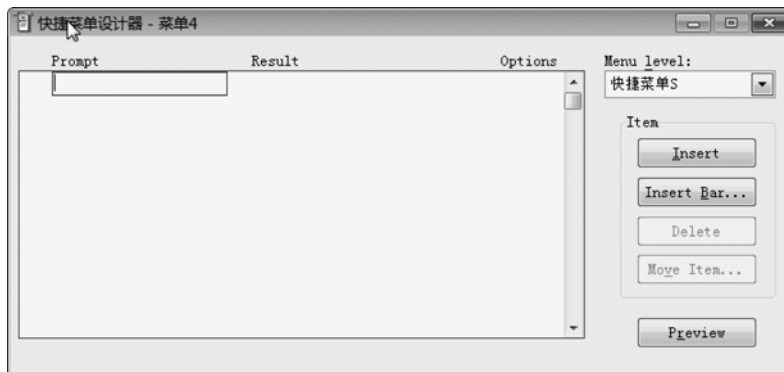


图 7-7 “快捷菜单设计器”窗口

(4) 与下拉式菜单相似, 在“快捷菜单设计器”窗口设计快捷菜单, 生成相应的菜单程序文件。

(5) 在快捷菜单的“清理”代码中添加清除菜单命令, 使得在选择、执行菜单命令后能够及时清除菜单, 释放相应的内存空间。

命令格式:

Release popup <快捷菜单名>[extended]

(6) 在表单设计器下, 选中要添加快捷菜单的对象。在选定对象的 RightClick 事件中添加调用快捷菜单程序命令。

命令格式:

Do <快捷菜单程序文件名>

注意文件名的扩展名不能省略。

## 任务 7.2 实例训练

### 7.2.1 下拉式菜单设计实例

【例 7-1】建立一个名为 main 的菜单文件, 其主菜单包含“学生管理(S)”、“编辑(E)”、“退出(Q)”3 个菜单项, 它们的结果分别是激活弹出式菜单 menu1、激活弹出式菜单 menu2、将系统菜单恢复为标准设置。

menu1 菜单项包括: 学生信息、课程信息和学生成绩。

menu2 菜单项包括: 复制、剪切和粘贴 3 个选项, 它们分别调用相应的系统标准功能。操作步骤如下。

(1) 在命令窗口输入命令 MODIFY MENU main, 打开菜单设计器。

(2) 在菜单设计器中定义主菜单中菜单项的名称, 并设置菜单项的访问键, 如图 7-8 所示。

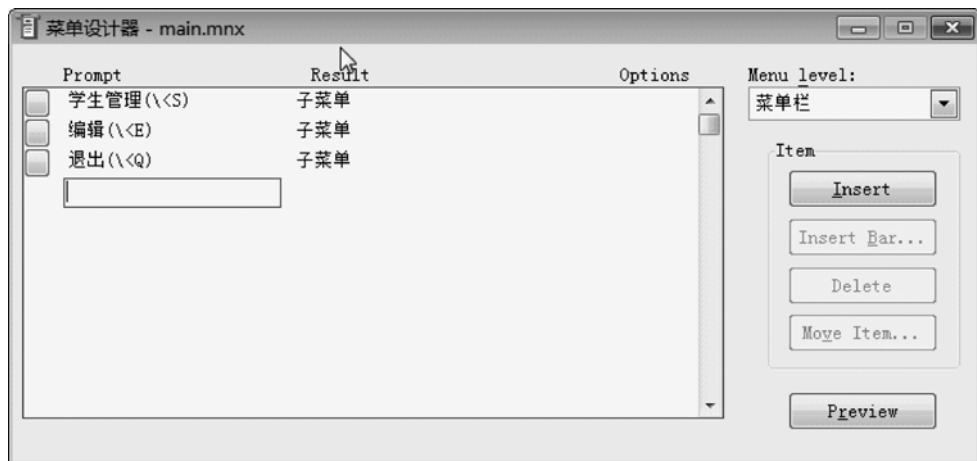


图 7-8 在菜单设计器中定义名称并设置访问键

(3) 为“退出”定义过程代码, 在“Result (结果)”列的下拉菜单中选择“过程”, 单

击 “Result (结果)” 列上的 “创建” 按钮，弹出过程编辑窗口，输入过程代码。

```
set sysmenu nosave
set sysmenu to default
```

(4) 定义弹出式菜单 menu1。

单击 “学生管理” 菜单项 “Result (结果)” 列上的 “创建” 按钮，使设计器窗口切换到子菜单页，然后设置各菜单项，在菜单设计器中定义 menu1 如图 7-9 所示。

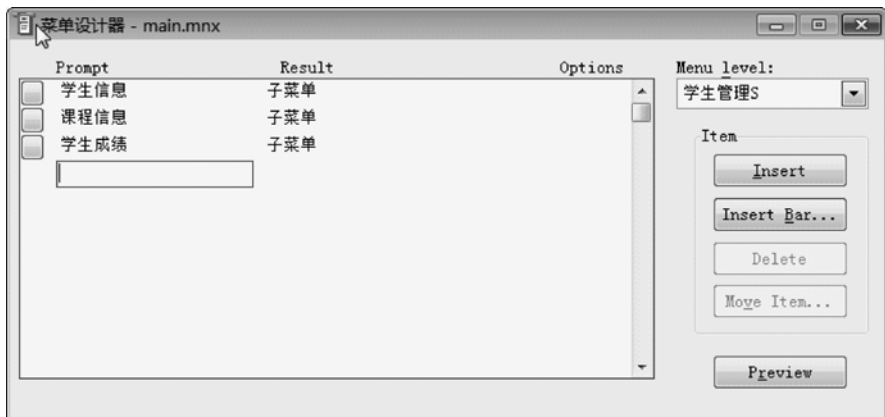


图 7-9 在菜单设计器中定义 menu1

(5) 定义弹出式菜单 menu2。

单击 “编辑” 菜单项 “结果” 列上的 “创建” 按钮，使设计器窗口切换到子菜单页，单击 “插入栏” 按钮，在弹出的对话框中分别选择复制、剪切和粘贴选项，“插入系统菜单栏” 对话框如图 7-10 所示。



图 7-10 “插入系统菜单栏” 对话框

“编辑” 菜单显示得到的结果，如图 7-11 所示。

(6) 设置弹出式菜单的内部名称。

选择 “显示” | “菜单选项” 命令，打开 “菜单选项” 对话框，如图 7-12 所示，然后在 “名称” 文本框中输入 menu1 或 menu2。

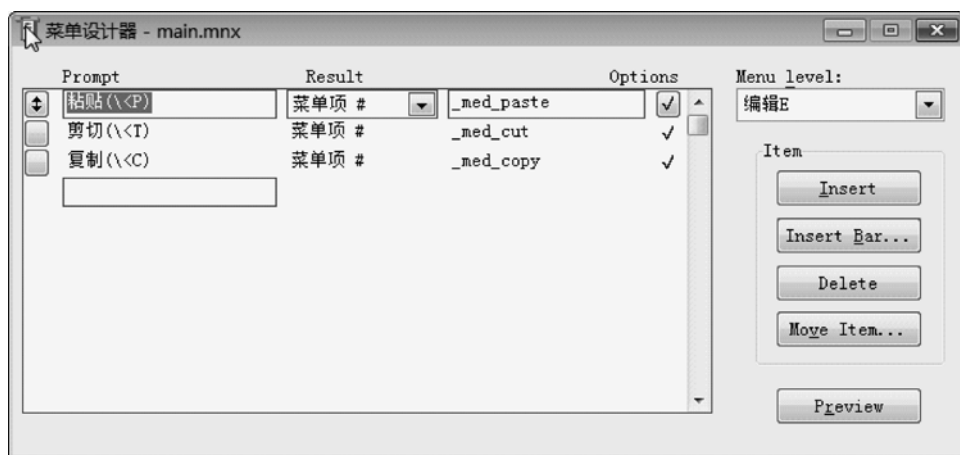


图 7-11 “编辑”菜单

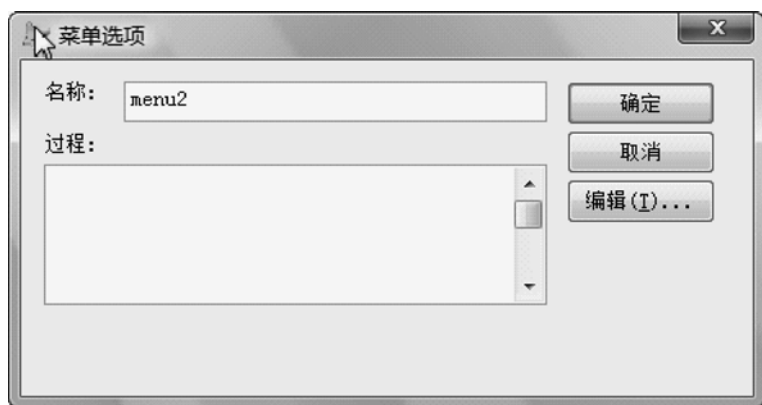


图 7-12 “菜单选项”对话框

(7) 相应的命令完成后，返回到“菜单栏”级别，选择“菜单”|“生成”命令，生成菜单的程序文件为 main.mpr。

### 7.2.2 快捷菜单设计实例

【例 7-2】给编辑框控件添加一个快捷菜单。

操作步骤如下。

(1) 选择“文件”|“新建”命令，在“新建”对话框中选中“菜单”单选按钮，然后单击“新建”按钮。

(2) 在“新建菜单”对话框中单击“快捷菜单”按钮，进入“快捷菜单设计器”窗口，如图 7-13 所示，在其中进行设计。

(3) 选择“菜单”|“生成”命令，产生菜单的程序文件为 main1.mpr。

(4) 创建一个包含编辑框控件的表单，定义编辑框控件的 RightClick 属性为一个过程。过程代码为：

```
DO f:\学生学籍管理系统\MENUS\MAIN1.MPR
```



图 7-13 “快捷菜单设计器”窗口

注意：当 main1.mpr 与我们所创建的表单不在同一个文件夹里面时，需要写明菜单的详细目录。例如，此处 main1.mpr 在 f:\学生学籍管理系统\MENUS 文件夹中，而我们建立的表单“快捷菜单运行.scx”在 f:\学生学籍管理系统文件夹中，就必须将编辑框控件的 RightClick 属性的过程代码写成 DO f:\学生学籍管理系统\MENUS\MAIN1.MPR。

(5) 运行表单，可能得到如图 7-14 所示的结果。



图 7-14 快捷菜单运行结果

### 7.2.3 给菜单指定菜单任务

菜单选项设计完成后，还要给每个菜单选项指定任务，菜单的设计工作才算完成。菜单选项的任务，可以是子菜单、命令或程序。

【例 7-3】 给主菜单“学生管理”的各子菜单指定任务。

操作步骤如下。

(1) 在 Visual FoxPro 系统的主菜单下，打开菜单文件 main.mnx，进入“菜单设计器”窗口。

(2) 在“菜单设计器”窗口中选择“学生管理”菜单项，然后单击“编辑”按钮，进入“菜单设计器”子菜单操作窗口。

(3) 将“结果”列设为“命令”，如图 7-15 所示。

为“学生信息”菜单项输入命令 do form student。

为“学生成绩”菜单项输入命令 do form result。

前提是表单 student 与 result 都存在。在“菜单设计器”窗口中保存设置并退出。

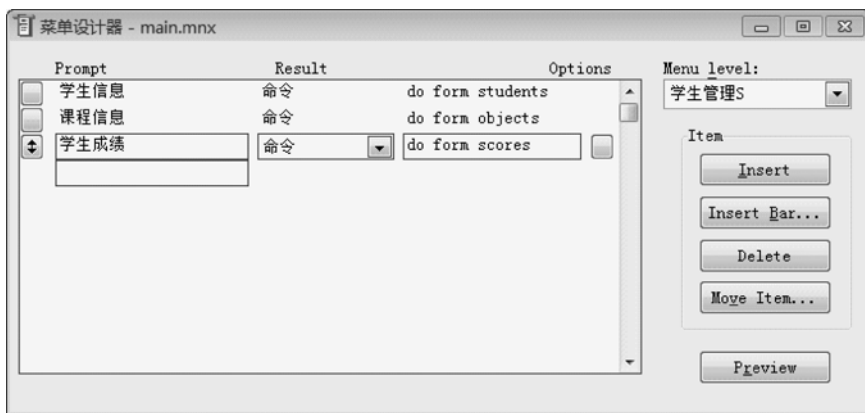


图 7-15 将“结果”列设为“命令”

#### 7.2.4 运行菜单

完成相应的设计工作后，用户可以通过以下两种方式来运行菜单。

(1) 命令方式。

命令格式为：do <菜单文件名>

例如，编写程序运行 main.mpr，操作步骤如下。

在 Visual FoxPro 系统下新建 main.prg 程序。

在程序窗口中输入代码。

```
_screen.caption="学生学籍管理"
set sysmenu on
do main.mpr
```

保存文件。

(2) 在 Visual FoxPro 系统下运行菜单文件。

操作步骤如下。

在 Visual FoxPro 系统主菜单下，选择“程序”|“运行”命令，进入运行窗口。

选中要运行的菜单文件，例如，main.mpr，单击“运行”即可，运行菜单后的结果如图 7-16 所示。



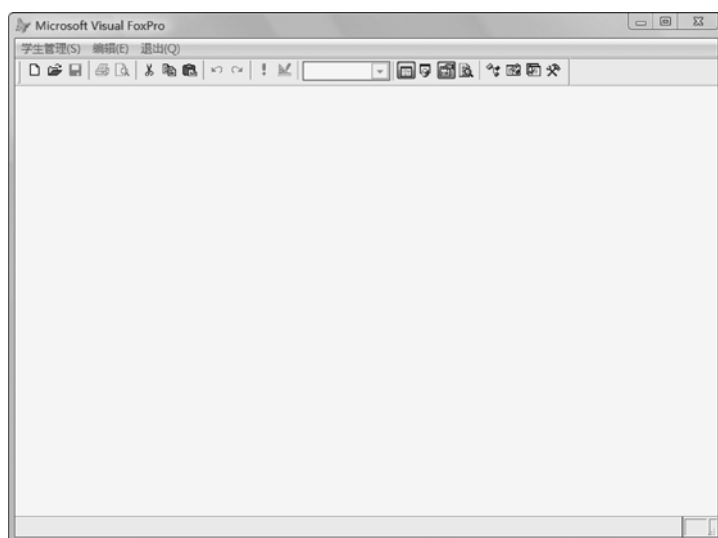


图 7-16 运行菜单后的结果

## 任务 7.3 项目分析

在这个项目中，我们完成了“学生学籍管理系统”主菜单的设计，包括“学生管理”、“编辑”和“退出”3个选项，每个选项又有自己的子菜单，我们分别来看它们的结构。

### 7.3.1 系统主菜单

系统主菜单如图 7-17 所示。



图 7-17 系统主菜单

### 7.3.2 “学生管理”菜单

“学生管理”菜单如图 7-18 所示。

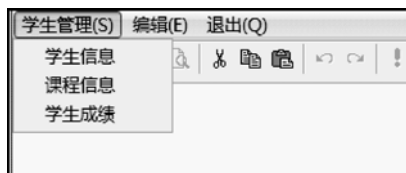


图 7-18 “学生管理”菜单

### 7.3.3 “编辑”菜单

“编辑”菜单如图 7-19 所示。

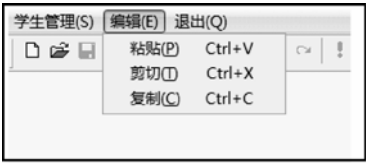


图 7-19 “编辑”菜单

## 任务 7.4 模仿设计与自主设计

### 7.4.1 建立常规菜单

建立菜单 main，如表 7-1 所示。

表 7-1 菜单 main

| 数据编辑 | 数据统计 | 信息查询 | 打印输出 | 退出系统 |
|------|------|------|------|------|
|      |      |      |      |      |
|      |      |      |      |      |

(1) 为“退出系统”菜单输入命令：set sysmenu to default

(2) 在上面菜单的基础上创建下拉子菜单。

数据编辑的下拉子菜单：录入、修改、删除。

数据统计的下拉子菜单：学生统计、成绩统计。

信息查询的下拉子菜单：按编号查询、按姓名查询、按性别查询、按年龄查询、按职称查询、按部门查询。

打印输出的下拉子菜单：课程成绩打印、个人成绩打印。

(3) 菜单项分组，如表 7-2 所示。

表 7-2 菜单项分组

| 数据编辑 | 数据统计 | 信息查询  | 打印输出   | 退出系统 |
|------|------|-------|--------|------|
| 录入   | 学生统计 | 按编号查询 | 课程成绩打印 |      |
| 修改   | 成绩统计 | 按姓名查询 | 个人成绩打印 |      |
| 删除   |      | 按性别查询 |        |      |
|      |      | 按年龄查询 |        |      |
|      |      | 按职称查询 |        |      |
|      |      | 按部门查询 |        |      |

(4) 添加热键及快捷键，如表 7-3 所示。

表 7-3 添加热键及快捷键

| 数据编辑(E) | 数据统计(T) | 信息查询(T)      | 打印输出(P) | 退出系统(X) |
|---------|---------|--------------|---------|---------|
| 录入      | 学生统计    | 按编号查询 Ctrl+B | 课程成绩打印  |         |
| 修改      | 成绩统计    | 按姓名查询 Ctrl+X | 个人成绩打印  |         |
| 删除      |         | 按性别查询 Ctrl+S |         |         |
|         |         | 按年龄查询 Ctrl+A |         |         |
|         |         | 按职称查询 Ctrl+Z |         |         |
|         |         | 按部门查询 Ctrl+D |         |         |



(5) 在状态栏显示信息。

单击“数据编辑”菜单时，状态栏显示“可以对数据进行录入、修改和删除操作”。

(6) 生成菜单。选择“菜单”|“生成”命令。

(7) 运行菜单。

打开项目管理器，选择“其他”|“菜单文件”|“运行”命令。

命令窗口：do 菜单程序名.mpr

通过菜单运行菜单程序。

#### 7.4.2 建立快捷菜单

建立快捷菜单 KJ。

(1) 新建数据库 create database。

(2) 打开数据库 open database。

(3) 运行快捷菜单。

#### 7.4.3 菜单综合设计

(1) 打开菜单 main，设置属性。

选择“显示”|“常规选项”命令，在“常规选项”对话框中选中“顶层表单”复选框。

(2) 建立一个名为 xsxt 的表单，设置属性。

CAPTION=学生管理系统

SHOWWINDOWS=2 (作为顶层表单)

WINDOWSSTATE=2 (最大化)

表单 Init 事件为：do main.mpr with this, .t.

on key label rightmouse do kj.mpr

Destroy 事件：clear event

Close DataBase

(3) 退出系统命令改为：quit

(4) 运行表单，观察菜单位置，单击鼠标右键，调用快捷菜单。

### 项目小结 7

项目 7 介绍了 Visual FoxPro 菜单设计 (包括下拉式菜单和快捷菜单) 的基本方法，以及如何指定菜单任务和运行菜单等问题。

### 项目习题 7

#### 1. 选择题

(1) 在 VFP 中，有关菜单的下列说法不正确的是 ( )。

A. 执行菜单源程序文件的命令为“DO MENU 菜单文件名”

B. 在菜单设计器中设计菜单时, 要给菜单项分组, 可在相应分组位置加入 “\ - ”

C. 在菜单设计器中设计菜单时, 要给菜单项设置访问键 F, 可在该菜单项后加入 “(\<F) ”

D. 在 VFP 中, 也可以设计快捷菜单

(2) 在菜单设计器中设计好菜单之后, 系统即生成一个扩展名为 ( ) 的菜单文件, 它不是一个可执行的程序。

A. .MNX          B. .MPR          C. .QPR          D. .DBC

(3) 在 Visual FoxPro 中, 使用菜单设计器定义菜单, 最后生成的菜单程序的扩展名是 ( )。

A. .MNX          B. .PRG          C. .MPR          D. .SPR

(4) 为表单建立了快捷菜单 MYMENU, 调用快捷菜单的命令代码 DO mymenu.mpr WITH THIS 应该放在表单的 ( ) 事件中。

A. Desory 事件    B. Init 事件    C. Load 事件    D. RightClick 事件

(5) 以下是与设置系统菜单有关的命令, 其中错误的是 ( )。

A. SET SYSMENU DEFAULT    B. SET SYSMENU TO DEFAULT  
C. SET SYSMENU NOSAVE    D. SET SYSMENU SAVE

(6) 在 Visual FoxPro 中, 要运行菜单文件 menu1.mpr, 可以使用命令 ( )。

A. DO menu1                      B. DO menu1.mpr  
C. DO MENU menu1                D. RUN menu1

## 2. 填空题

(1) 为了从用户菜单返回默认的系统菜单, 应该使用命令 SET \_\_\_\_\_ TO DEFAULT。

(2) 要将一个弹出式菜单作为某个控件的快捷菜单, 通常是在该控件的 \_\_\_\_\_ 事件代码中添加调用弹出式菜单程序的命令。

(3) 典型的菜单系统一般是一个下拉式菜单, 下拉式菜单通常由一个 \_\_\_\_\_ 和 \_\_\_\_\_ 组成。

(4) 要为表单设计下拉式菜单, 首先要在菜单设计时, 在 \_\_\_\_\_ 对话框中选中 “顶层表单” 复选框, 其次要将表单的 \_\_\_\_\_ 属性值设置为 2, 使其成为顶层表单, 最后要在 \_\_\_\_\_ 事件代码中设置调用菜单程序的命令。



# 项目 8

## 设计学生学籍管理 系统报表

### 项目要点

使用报表向导、报表设计器创建报表，修改报表布局、美化报表的方法。

### 学习目标

了解报表的基础知识，设计报表、打印报表的基本方法。

掌握利用向导制作报表及快速制作报表方法、报表设计器的功能及修改、美化报表的方法、报表控件的基本功能及添加方法。

对数据库进行操作时，数据和文档的输出方式通常有两种：屏幕显示和打印机打印。因为屏幕显示受到屏幕尺寸和不能永久保存的限制，所以打印机打印就成为数据库数据及文档输出不可替代的手段。报表就是一种可以通过打印机输出数据库数据及文档的实用方式。本项目将结合实例来介绍如何创建、修改和打印报表。

## 任务 8.1 熟悉报表基础知识

### 8.1.1 报表概述

报表主要包括两部分内容：数据源和布局。数据源是报表的数据来源，报表的数据源通常是数据库中的表或自由表，也可以是视图、查询或临时表。视图和查询是对数据库中

的数据进行筛选、排序和分组，在定义了一个表、一个视图或查询之后，便可以创建报表。

### 8.1.2 利用报表向导创建报表

使用报表向导首先要打开报表的数据源。数据源可以是数据库表，也可以是视图或临时表。

启用报表向导有以下 3 种途径。

(1) 打开项目管理器，选择“文档”选项卡，从中选择“报表”选项，然后单击“新建”按钮，在弹出的“新建报表”对话框中单击“报表向导”按钮，如图 8-1 所示。

(2) 选择“文件”|“新建”命令，或者单击“常用”工具栏中的“新建”按钮，打开“新建”对话框，在文件类型栏中选择“报表”，然后单击“向导”按钮。

(3) 选择“工具”|“向导”|“报表”命令。

报表启动时，首先弹出“Wizard Selection (向导选取)”对话框，如图 8-2 所示。



图 8-1 “新建报表”对话框



图 8-2 “Wizard Selection (向导选取)”对话框

如果数据源是一个表，则应选取“Report Wizard (报表向导)”；如果数据源包括多个表，则应选取“One-to-Many Wizard (一对多报表向导)”。

### 8.1.3 利用报表设计器创建报表

Visual FoxPro 提供的报表设计器允许用户通过直观的操作来直接设计报表或修改报表。

#### 1. 打开报表设计器

调用方法有以下 3 种。

(1) 在项目管理器下调用。

选择“文档”选项卡，从中选择“报表”选项，然后单击“新建”按钮，在弹出的“新建报表”对话框中单击“新建报表”按钮。

(2) 选择“文件”|“新建”命令，或者单击“常用”工具栏中的“新建”按钮，打开“新建”对话框，在文件类型栏中选择“报表”，然后单击“新建文件”按钮。



### (3) 命令格式。

Create report [<报表文件名>]

报表设计器如图 8-3 所示。

## 2. “报表设计器”工具栏

当打开“报表设计器”窗口后,就会出现“报表设计器”工具栏,如图 8-4 所示。

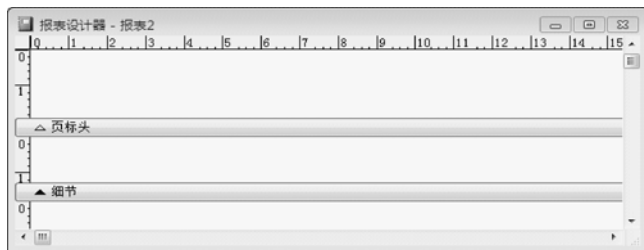


图 8-3 报表设计器

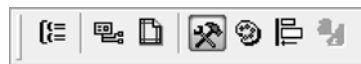









图 8-4 “报表设计器”工具栏

其中各按钮的功能如下。

- (1)“数据分组”按钮,显示“数据分组”对话框,用于创建数据分组及指定其属性。
- (2)“数据环境”按钮,显示报表的“数据环境设计器”窗口。
- (3)“页面设置”按钮,显示报表的“报表属性”窗口。
- (4)“报表控件工具栏”按钮,显示或关闭“报表控件”工具栏。
- (5)“调色板工具栏”按钮,显示或关闭“调色板”工具栏。
- (6)“布局工具栏”按钮,显示或关闭报表工具栏。
- (7)“字体属性”按钮,显示报表的“字体属性”窗口。

## 3. “报表控件”工具栏

“报表控件”工具栏如图 8-5 所示。

主要按钮及其功能如下。









- (1)“选择对象”按钮,移动控件或更改控件的大小。
- (2)“标签”按钮,在报表上创建一个标签控件,用于输入并显示与记录无关的数据。
- (3)“域控件”按钮,在报表上创建一个字段控件,用于显示字段、内存变量或其他表达式的内容。
- (4)“线条”按钮,“矩形”按钮,“圆角矩形”按钮分别用于绘制相应的图形。
- (5)“图片/OLE 绑定控件”按钮,显示图片或通用型字段内容。
- (6)“按钮锁定”按钮,允许添加多个相同类型的控件,而不需要多次选中该控件按钮。



图 8-5 “报表控件”工具栏

## 8.1.4 打印报表

如果要在程序中打印报表或在命令窗口中打印报表,可使用下面的命令行:

REPORT FORM 报表文件名 [范围]  
[FOR 条件] [WHILE 条件] [NOEJECT]  
[NOCONSOLE] TO PRINTER

该命令的功能是按照报表文件的设计安排，把由[范围]和 FOR、WHILE 子句筛选出来的记录以数据报表的形式打印在纸张上。

[范围] 可以是 ALL、NEXT n、RECORD n、REST。

[NOEJECT] 表示打印前不换页。

[NOCONSOLE] 表示打印时打印的内容不在屏幕上显示。

## 任务 8.2 实例训练

### 8.2.1 利用报表向导创建报表实例

【例 8-1】利用报表向导，对 students 表创建报表。

操作步骤如下。

- (1) 打开数据库文件“学籍管理数据库.dbc”。
- (2) 选择“文件”|“新建”命令，或者单击“常用”工具栏中的“新建”按钮，打开“新建”对话框，在文件类型栏中选择“报表”，然后单击“向导”按钮。
- (3) 在“向导选取”对话框中选取“报表向导”，单击“确定”按钮。
- (4) 在“Selected Field (字段选取)”对话框中选取相应的表与字段，如图 8-6 所示。



图 8-6 “Selected Field (字段选取)”对话框

(5) 选择好相关字段后，单击“Next (下一步)”按钮，弹出“Group Record (分组记录)”对话框，选择以“学号”分组，如图 8-7 所示。

(6) 单击“Next (下一步)”按钮，弹出“Choose Report Style (选择报表样式)”对话框，如图 8-8 所示。

(7) 选择好相应的报表样式后，单击“Next (下一步)”按钮，弹出“Define Report Layout (定义报表布局)”对话框，如图 8-9 所示。



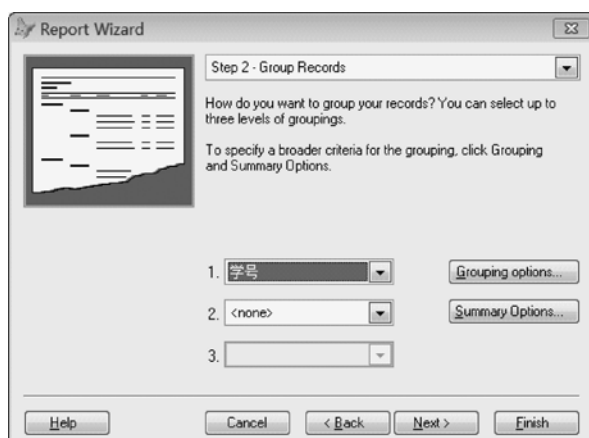


图 8-7 “Group Record (分组记录)”对话框

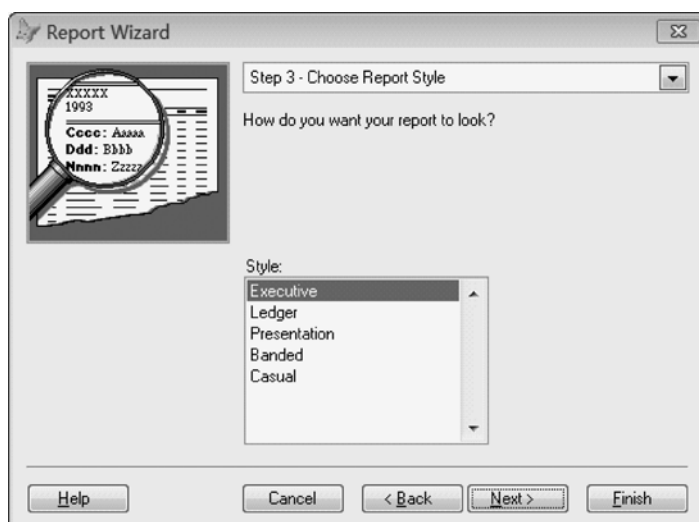


图 8-8 “Choose Report Style (选择报表样式)”对话框

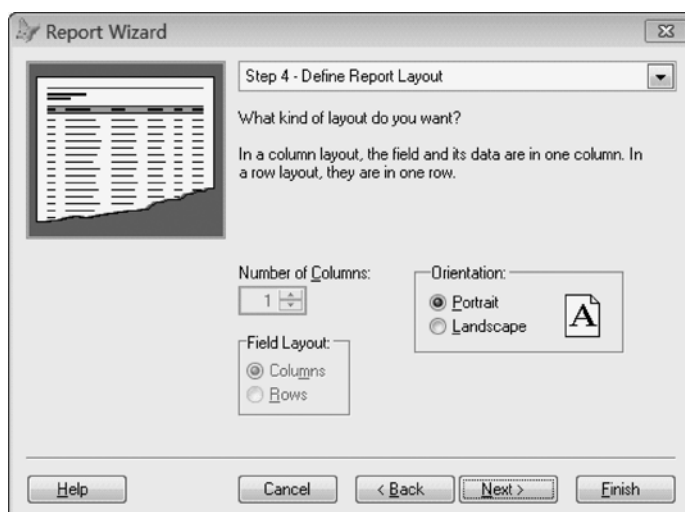


图 8-9 “Define Report Layout (定义报表布局)”对话框

(8) 单击“Next (下一步)”按钮,弹出“Sort Record (排序记录)”对话框,选择相应的字段进行排序,选择按“姓名”排序,如图 8-10 所示。



图 8-10 “Sort Record (排序记录)”对话框

(9) 单击“Next (下一步)”按钮,确定报表的标题,完成最后一步的设置,“Finish (完成)”对话框如图 8-11 所示。

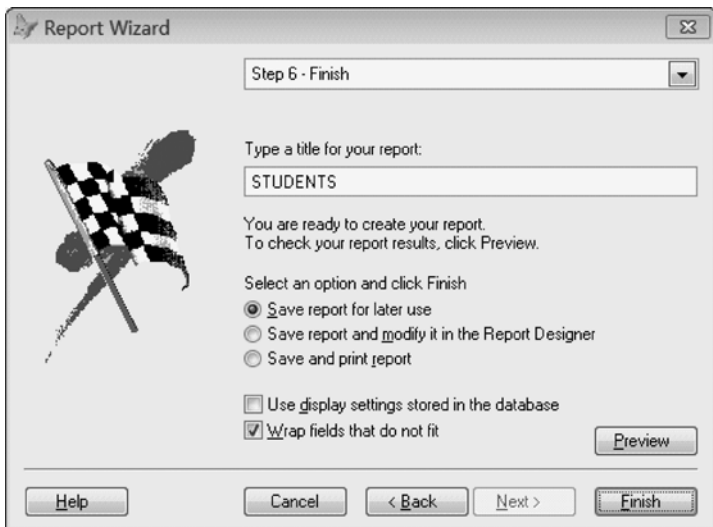


图 8-11 “Finish (完成)”对话框

(10) 单击“Finish (完成)”按钮,进行保存,命名为 students.frx。

### 8.2.2 利用报表设计器创建报表实例

【例 8-2】以 students.dbf 为数据源利用报表设计器来创建报表文件。  
操作步骤如下。

(1) 打开数据库“学籍管理数据库.dbc”。



(2) 选择“文件”|“新建”命令,或者单击“常用”工具栏的“新建”按钮,打开“新建对话框”,在文件类型栏中选择“报表”,然后单击“新建”按钮。

(3) 在报表设计器中单击鼠标右键,选择“数据环境”命令,打开数据环境设计器窗口,在此处点击右键,选择“添加”命令,添加作为报表数据源的表 students.dbf,添加表后的数据环境设计器如图 8-12 所示。



图 8-12 添加表后的数据环境设计器

(4) 关闭数据环境设计器窗口,回到 Visual FoxPro 系统的主菜单下。

(5) 选择“报表”|“快速报表”命令,弹出“快速报表”对话框,如图 8-13 所示。



图 8-13 “快速报表”对话框

(6) 在“快速报表”对话框中,单击“字段”按钮,弹出“字段选择器”对话框,如图 8-14 所示。



图 8-14 “字段选择器”对话框

(7) 在“字段选择器”对话框中选择出现在报表中的字段，然后单击“确定”按钮，返回到“快速报表”对话框，再单击“确定”按钮，返回“报表设计器”窗口，如图 8-15 所示。



图 8-15 “报表设计器”窗口

(8) 选择“报表”|“可选区段”命令，打开“Report Properties (报表属性)”对话框，如图 8-16 所示。

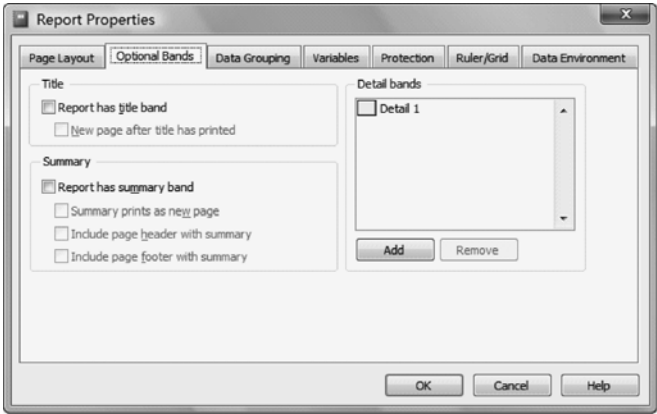


图 8-16 “Report Properties (报表属性)”对话框

(9) 选中“Report has title band (报表有标题带区)”复选框，并单击“OK (确定)”按钮，回到“报表设计器”窗口，如图 8-17 所示。



图 8-17 “报表设计器”窗口

(10) 在标题区，利用标签控件为报表添加一个标题“学生学籍信息报表”，如图 8-18 所示。

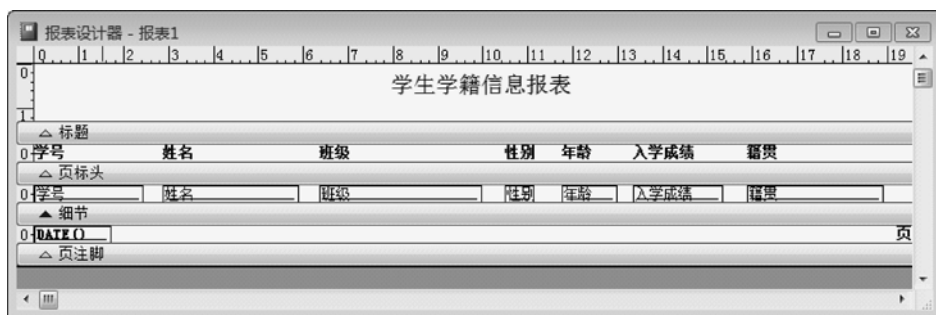


图 8-18 添加标题“学生学籍信息报表”

- (11) 在“报表设计器”中选择“显示”|“预览”命令，进入报表设计器的预览窗口。
- (12) 当对报表的格式满意后，退出报表设计器，系统提示是否要保存报表。
- (13) 单击“是”按钮，输入报表名称，如 report1，进行保存。

### 8.2.3 修改报表

如果发现报表有不尽如人意的地方，可以对报表进行修改。

例如，对上节产生的报表 report1 进行修改，操作步骤如下。

- (1) 在 Visual FoxPro 系统的主菜单下，打开 report1，进入“报表设计器”窗口。
- (2) 在“报表设计器”窗口中，打开“报表”菜单，修改相应的数据或布局。
- (3) 修改完毕后，可预览，再次进行保存。

## 任务 8.3 项目分析

### 8.3.1 students 报表

students 报表如图 8-19 所示。

| STUDENTS |             |         |    |    |      |
|----------|-------------|---------|----|----|------|
| 03/19/11 |             |         |    |    |      |
| 学号       | 姓名<br>籍贯    | 班级      | 性别 | 年龄 | 入学成绩 |
| 0041001  |             |         |    |    |      |
|          | 洪明信<br>广东信宜 | 09高职多媒体 | 男  | 23 | 320  |
| 0041002  |             |         |    |    |      |
|          | 李小华<br>广东揭阳 | 09高职多媒体 | 女  | 23 | 310  |
| 0041003  |             |         |    |    |      |
|          | 张晶林<br>河南郑州 | 09高职计信管 | 男  | 22 | 310  |
| 0104001  |             |         |    |    |      |
|          | 陈达文<br>新疆奎屯 | 09高职计信管 | 男  | 23 | 345  |
| 0104002  |             |         |    |    |      |

图 8-19 students 报表

### 8.3.2 学生学籍信息报表

学生学籍信息报表如图 8-20 所示。

| 学号      | 姓名  | 班级      | 性别 | 年龄 | 入学成绩 | 籍贯   |
|---------|-----|---------|----|----|------|------|
| 0104001 | 陈达文 | 09高职计信管 | 男  | 23 | 345  | 新疆奎屯 |
| 0104002 | 陈桂芬 | 09高职计信管 | 女  | 18 | 240  | 新疆伊犁 |
| 0104003 | 江培强 | 09高职计信管 | 男  | 22 | 420  | 广东高州 |
| 0104004 | 林江文 | 09高职计信管 | 男  | 22 | 380  | 广东湛江 |
| 0104005 | 林世鑫 | 09高职计信管 | 男  | 26 | 290  | 广东徐闻 |
| 0104006 | 陈玉强 | 09高职计信管 | 男  | 22 | 315  | 广东湛江 |
| 0104007 | 谭旭金 | 09高职计信管 | 男  | 22 | 300  | 广东廉江 |
| 0041001 | 洪明信 | 09高职多媒体 | 男  | 23 | 320  | 广东信宜 |
| 0041002 | 李中华 | 09高职多媒体 | 女  | 23 | 310  | 广东揭阳 |
| 0041003 | 张晶林 | 09高职计信管 | 男  | 22 | 310  | 河南郑州 |

图 8-20 学生学籍信息报表

### 8.3.3 学生成绩报表

根据之前的项目中建立的学生信息与成绩视图创建学生成绩报表，如图 8-21 所示。

| 学号             | 姓名  | 班级      | 性别 | 年龄 | 课程编号   | 成绩 |
|----------------|-----|---------|----|----|--------|----|
| <b>0104001</b> |     |         |    |    |        |    |
|                | 陈达文 | 09高职计信管 | 男  | 23 | jsj001 | 80 |
|                | 陈达文 | 09高职计信管 | 男  | 23 | jsj002 | 76 |
|                | 陈达文 | 09高职计信管 | 男  | 23 | jsj005 | 82 |
| <b>0104002</b> |     |         |    |    |        |    |
|                | 陈桂芬 | 09高职计信管 | 女  | 18 | jsj003 | 90 |
|                | 陈桂芬 | 09高职计信管 | 女  | 18 | jsj006 | 86 |
| <b>0104004</b> |     |         |    |    |        |    |
|                | 林江文 | 09高职计信管 | 男  | 22 | jsj007 | 95 |
| <b>0104005</b> |     |         |    |    |        |    |
|                | 林世鑫 | 09高职计信管 | 男  | 26 | jsj001 | 83 |
| <b>0041001</b> |     |         |    |    |        |    |
|                | 洪明信 | 09高职多媒体 | 男  | 23 | jsj002 | 91 |
|                | 洪明信 | 09高职多媒体 | 男  | 23 | jsj003 | 86 |

图 8-21 学生成绩报表



## 任务 8.4 模仿设计与自主设计

### 8.4.1 创建“图书信息报表”

要求如下。

- (1) 利用报表向导创建“图书信息报表”。
- (2) 该报表显示图书编号、图书名称、作者、出版社、出版日期、价格和数量信息。

### 8.4.2 创建快速报表“租借信息报表”

要求如下。

- (1) 首先根据“图书信息表”、“租借者信息表”、“租借信息表”创建一个“租借信息视图”，然后根据该视图创建“租借信息报表”。
- (2) 利用报表设计器创建该报表。
- (3) 为该报表添加标题“租借信息报表”，并设置字体和大小。
- (4) 为报表添加页脚注，显示制作人信息。

## 项目小结 8

项目 8 介绍了在 Visual FoxPro 中如何用向导和设计器来创建报表，以及报表的修改和运行等问题。

## 项目习题 8

### 1. 选择题

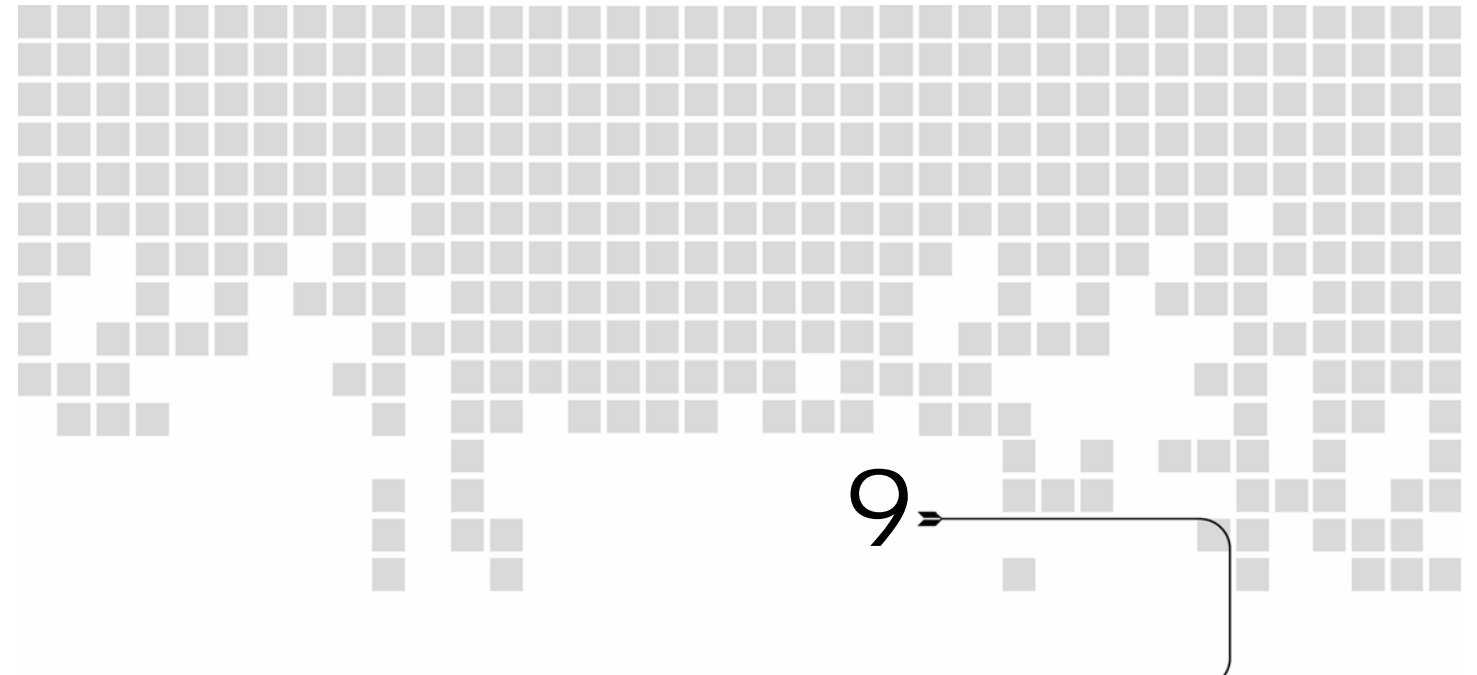
- (1) 报表的数据源可以是 ( )。  
A. 自由表或其他报表                      B. 数据库表、自由表和视图  
C. 数据库表、自由表和查询              D. 表、查询和视图
- (2) 在报表设计器中可以使用的控件是 ( )。  
A. 标签、域控件和线条                      B. 标签、域控件和列表框  
C. 标签、文本框和列表框                      D. 布局和数据源
- (3) 在创建快速报表时，基本带区包括 ( )。  
A. 标题、细节和总结                      B. 页标头、细节和页注脚  
C. 组标头、细节和页注脚                      D. 报表标题、细节和页注脚
- (4) 若要创建一个 3 级分组报表，第一个分组表达式是“部门”，第二个分组表达式是“性别”，第三个分组表达式是“基本工资”，当前索引的索引表达式应当是 ( )。

- A. 部门+性别+基本工资      B. 部门+性别+STR(基本工资)  
C. STR(基本工资)+性别+部门      D. 性别+部门+STR(基本工资)
- (5) 在 Visual FoxPro 中, 报表是由(    )和(    )两个基本部分组成的。
- A. 视图和布局      B. 数据库和布局  
C. 数据表和布局      D. 数据源和布局

## 2. 填空题

- (1) 设计报表通常包括两部分内容: \_\_\_\_\_ 和 \_\_\_\_\_。
- (2) “图片/Active 绑定控件”按钮用于显示\_\_\_\_\_和\_\_\_\_\_的内容。
- (3) 如果已对报表进行了数据分组, 报表会自动包含\_\_\_\_\_和\_\_\_\_\_带区。
- (4) 多栏报表的栏目数可以通过\_\_\_\_\_来设置。
- (5) 在命令窗口中可以使用\_\_\_\_\_命令来打印指定的报表。
- (6) 报表文件的扩展名是\_\_\_\_\_。





# 项目 9

## 公司人事管理系统

### 项目要点

通过一个公司人事管理系统的实例来说明创建一个完整的数据库应用系统所需要的工作步骤。

### 学习目标

了解应用程序设计的基本思想及应用程序的后续处理方法。  
掌握应用程序设计的开发步骤和模块化设计思想。

通过前面的学习，相信读者们已经掌握了 Visual FoxPro 9.0 的使用方法。在本项目中，我们将通过一个公司人事管理系统的设计来集中介绍应用程序设计的基本方法。

## 任务 9.1 系统规划

### 9.1.1 模块化设计

在开发一个应用程序前，设计人员应首先考虑整个项目的规划。一个人事管理系统应具有以下几个模块。

- (1) 数据库模块：整个系统的核心，所有人员信息必须保存在数据库的数据表中。
- (2) 登录验证模块：用来确认用户的有效性。
- (3) 主显模块：显示数据表的数据。
- (4) 查询模块：用来查询数据表的内容。
- (5) 报表模块：用来显示报表，便于打印。

- (6) 菜单模块：为了便于操作而制作的菜单。
- (7) 帮助说明模块：提供系统操作的基本方法说明。

### 9.1.2 系统流程图设计

由上述模块可知，整个系统的流程图如图 9-1 所示。

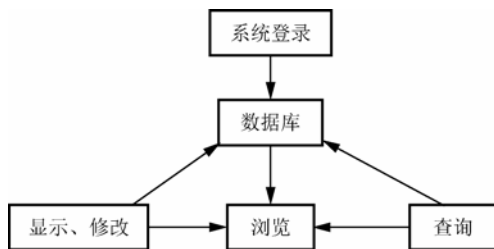


图 9-1 系统流程图

### 9.1.3 详细设计

根据所需要的模块和流程图，该系统的具体设计如下。

#### 1. 数据库

设计一个数据库，该库文件包括两个数据表，一个用来存储人员信息，另一个用来存储访问人员信息。

#### 2. 表单

系统包括以下几个表单。

- (1) 系统登录表单。
- (2) 浏览表单。
- (3) 逐条显示、修改表单。
- (4) 打印表单。
- (5) 查询表单。

#### 3. 其他辅助设计

辅助设计包括菜单和报表及帮助信息的制作。

## 任务 9.2 系统设计

在创建应用程序时，首先新建一个项目文件“人事管理系统.pjx”，然后在该项目管理器中，新建数据库“人员管理.dbc”。

### 9.2.1 创建数据表

在“人员管理.dbc”数据库文件中新建数据表 people.dbf 和 admin.dbf，并分别以“姓名”字段建立普通索引，people.dbf 的表结构和 admin.dbf 的表结构如图 9-2 和图 9-3 所示。



图 9-2 people.dbf 的表结构



图 9-3 admin.dbf 的表结构

然后分别在 people.dbf 数据表与 admin.dbf 数据表中添加若干条记录，people.dbf 和 admin.dbf 的记录信息如图 9-4 和图 9-5 所示。

| 姓名  | 性别 | 地址      | 电话      | 工资   | 年龄 | 备注 |
|-----|----|---------|---------|------|----|----|
| 张婉梦 | 男  | 山西临汾市   | 4092102 | 2800 | 26 |    |
| 李勇  | 男  | 湖南长沙市   | 2233530 | 3200 | 28 |    |
| 吴婉琼 | 女  | 山东青岛市   | 6268170 | 3300 | 29 |    |
| 卢国荣 | 男  | 江西江阴市   | 2763353 | 2080 | 22 |    |
| 孟芬芳 | 女  | 浙江杭州市   | 2233538 | 4500 | 32 |    |
| 吕爱国 | 男  | 上海南汇区   | 7343630 | 6800 | 35 |    |
| 邓超  | 男  | 河北唐山市   | 6633537 | 4300 | 33 |    |
| 徐爱秀 | 女  | 四川成都市   | 7333538 | 4500 | 36 |    |
| 赵明明 | 男  | 广西桂林市   | 8933536 | 3400 | 30 |    |
| 王强  | 男  | 新疆乌鲁木齐市 | 4533531 | 4300 | 36 |    |

图 9-4 people.dbf 记录信息



|  | 姓名      | 密码      | 备注    |
|--|---------|---------|-------|
|  | manager | 110380  | 经理    |
|  | admin   | 761119  | 管理员   |
|  | guest   | 1234567 | 一般访问者 |

图 9-5 admin.dbf 记录信息

## 9.2.2 创建表单

### 1. 登录表单

登录表单是为了限制用户登录而设置的，对于一般的访问用户，只能进行访问而不能进行更改。在项目管理器中建立该表单，保存为 login.scx，系统登录表单如图 9-6 所示。

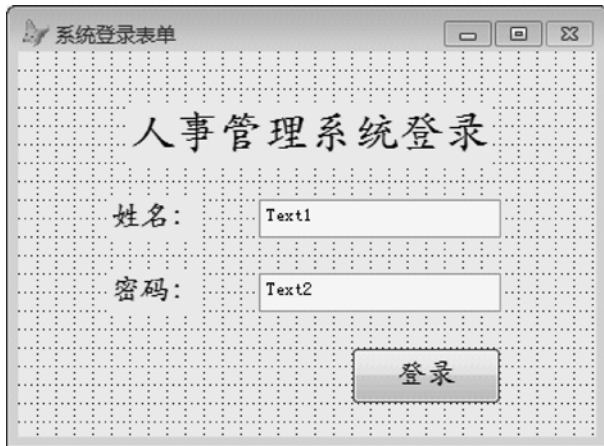


图 9-6 系统登录表单

给出各对象的事件代码。

该表单对象的 Load 事件代码为：

```
Open database f:\人事管理系统\人员管理.dbc
Use admin
Set order to 1
Go top
Set delete on &&设置数据表读取规则
```

该表单对象的 Destroy 事件代码为：

```
Use
Close database
Clear event
```

“登录”命令按钮的 Click 事件代码为：

```
Locate for (alltrim(姓名)==ALLTRIM(thisform.text1.value))
If found()
If ALLTRIM(thisform.text2.value)==alltrim(密码)
```



```

thisform.release
DO FORM display_search.scx &&正确后进入显示查询界面
Else
MessageBox("没有这个用户!",64,"!")
Endif
Endif

```

完成设置后，运行表单，如图 9-7 所示。



图 9-7 运行表单

## 2. 显示与查询表单

显示与查询表单是整个系统的关键，它把两个表单设置在一个表单内，分属不同的页框。操作步骤如下。

- (1) 新建一个表单，保存为 display\_search.scx。设置 Caption 属性为“人员管理表单”。
- (2) 在表单中添加一个页框控件 PageFrame1，设置 PageCount 属性为 2，设置 PageFrame1.page1 的 Caption 属性为“浏览”，PageFrame1.page2 的 Caption 属性为“查询”。
- (3) 为表单的数据环境设计器添加 people.dbf 数据表。
- (4) 用鼠标把数据表拖动到 Page1，添加页框控件，如图 9-8 所示。操作时应首先右击页框 PageFrame1，选择“编辑”选项，然后选择 Page1，再拖动数据表 people.dbf 到 Page1 内。

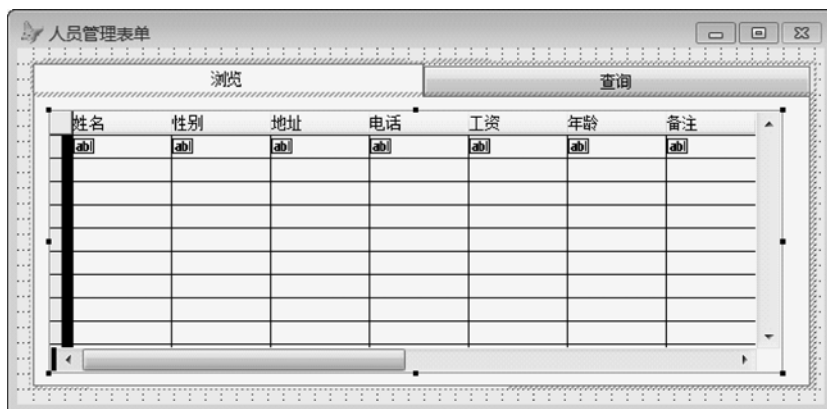


图 9-8 在表单中添加页框控件

(5) 选择 page2, 添加两个文本框控件、一个命令按钮和一个形状控件方框, 如图 9-9 所示。

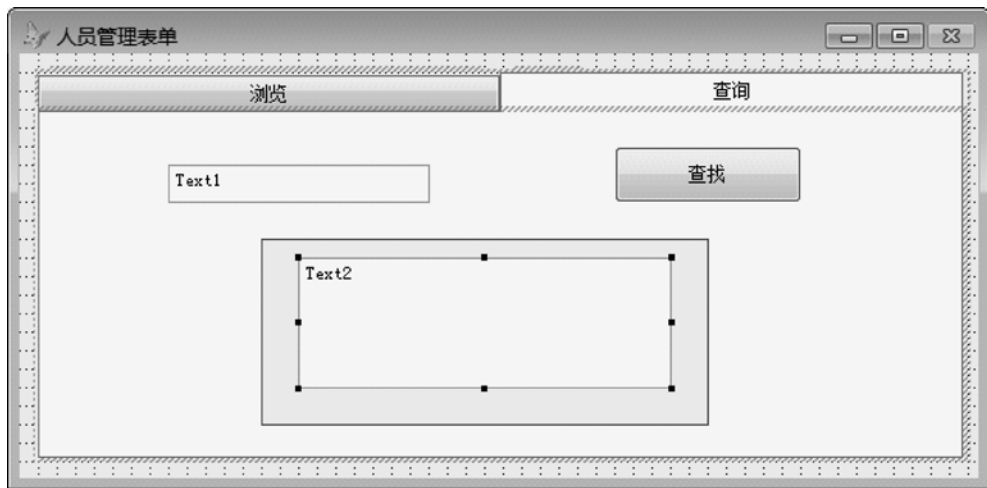


图 9-9 添加控件

(6) 向表单添加代码, 因为该表单为主表单, 需要添加菜单, 所以先向表单添加代码。该表单对象的 Load 事件代码为:

```
Open database f:\人事管理系统\人员管理.dbc
Use people
Set order to 1
Go top
Set delete on
```

页框对象中页面 Page2 的“查找”按钮 click 代码为:

```
Locate for (alltrim(姓名)==alltrim(thisform.pageframe1.page2.text1.value))
If found()
Thisform.pageframe1.page2.text2.value=alltrim(姓名)+alltrim(性别)+
alltrim(地址)+ alltrim(备注)
Else
MessageBox("没有找到该用户",64,"!")
endif
```

(7) 运行表单, 结果如图 9-10 和图 9-11 所示。

### 3. 显示、修改记录表单

设计一个表单 edit.scx, 用来提供记录的显示、修改等操作, 这里用表单向导来完成, 如图 9-12 所示。



| 姓名  | 性别 | 地址      | 电话      | 工资   | 年龄 | 备注 |
|-----|----|---------|---------|------|----|----|
| 邓超  | 男  | 河北唐山市   | 6633537 | 4300 | 33 |    |
| 李勇  | 男  | 湖南长沙市   | 2233530 | 3200 | 28 |    |
| 卢国荣 | 男  | 江西江阴市   | 2763353 | 2080 | 22 |    |
| 吕爱国 | 男  | 上海南汇区   | 7343630 | 6800 | 35 |    |
| 孟芬芳 | 女  | 浙江杭州市   | 2233538 | 4500 | 32 |    |
| 王殇  | 男  | 新疆乌鲁木齐市 | 4533531 | 4300 | 36 |    |
| 吴晓琼 | 女  | 山东青岛市   | 6268170 | 3300 | 29 |    |
| 徐爱秀 | 女  | 四川成都市   | 7333538 | 4500 | 36 |    |
| 张晓梦 | 男  | 山西临汾市   | 4092102 | 2800 | 26 |    |

图 9-10 运行结果 (1)

王殇

查找

王殇男新疆乌鲁木齐市

图 9-11 运行结果 (2)

edit

姓名: 姓名1

性别: 性别

地址: 地址1

电话: 电话1

工资: 工资1

年龄: 年龄1

备注: 备注1

Top Prev Next Bottom Find Print Add Edit Delete Exit

图 9-12 edit.scx 表单界面

#### 4. 打印表单

为了完成打印功能，特别设计一个表单 print.scx 以完成打印功能，如图 9-13 所示。

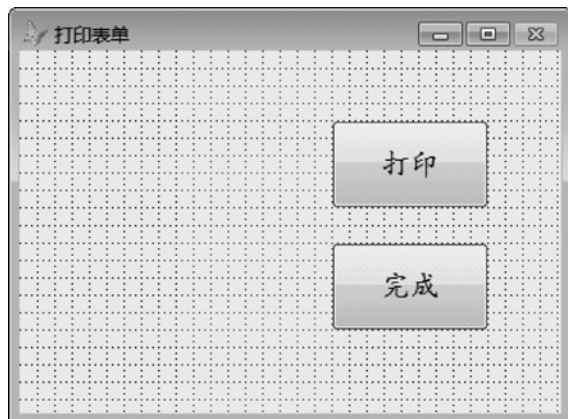


图 9-13 print.scx 表单

为“打印”按钮的 Click 事件编写如下代码：

```
Report form telebb noconsole to printer prompt
```

其中“telebb”是下一节要建立的报表。

为“完成”按钮的 Click 事件编写如下代码：

```
Thisform.release
```

#### 5. 帮助表单

在这里，我们通过一个帮助表单 help.scx 来对整个系统进行介绍，这也是设计一个系统所必须具备的部分。在表单中添加一个编辑框控件、一个按钮控件，并设置相关属性，如图 9-14 所示。

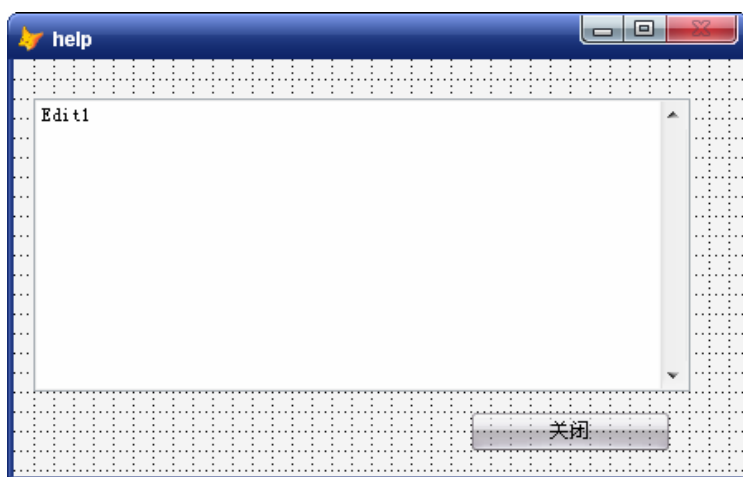


图 9-14 帮助表单 help.scx





在 Edit1 控件的 Init 事件里面添加如下代码：

```
Private t,tti
T=""
dimension tt(10)
Tt(1)="帮助"
Tt(2)="本软件为专业人事管理单位制作"
Tt(3)="本软件操作简单"
Tt(4)="为你提供最直接的帮助"
Tt(5)="快速帮助请按 F1"
Tt(6)="联系电话：010-6785432"
Tt(7)=""
Tt(8)="清竹制作室"
Tt(9)="2011 年"
For i=1 to 9 &&循环输出
 T=t+tt(i)+chr(13)
Endfor
This.value=t
```

为“关闭”按钮的 Click 事件编写如下代码：

```
thisform.Release
```

运行结果如图 9-15 所示。



图 9-15 help.scx 的运行结果

### 9.2.3 创建菜单和报表

#### 1. 创建菜单

在项目管理器中选择“菜单”选项，建立菜单 mainmenu.mnx。一级菜单如图 9-16 所示。

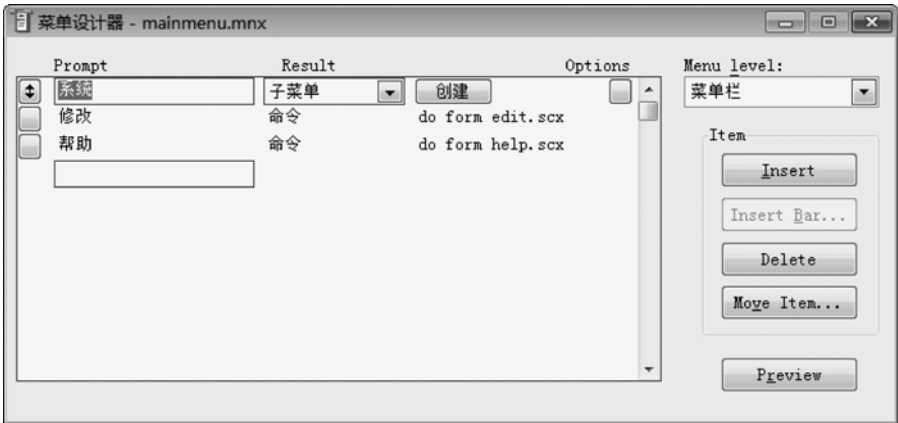


图 9-16 mainmenu 的一级菜单

在该级菜单中，“修改”和“帮助”为命令，分别为：do form edit.scx 和 do form help.scx。  
“系统”含有下拉菜单，单击右边的“创建”按钮，进入下级菜单的设置，如图 9-17 所示。

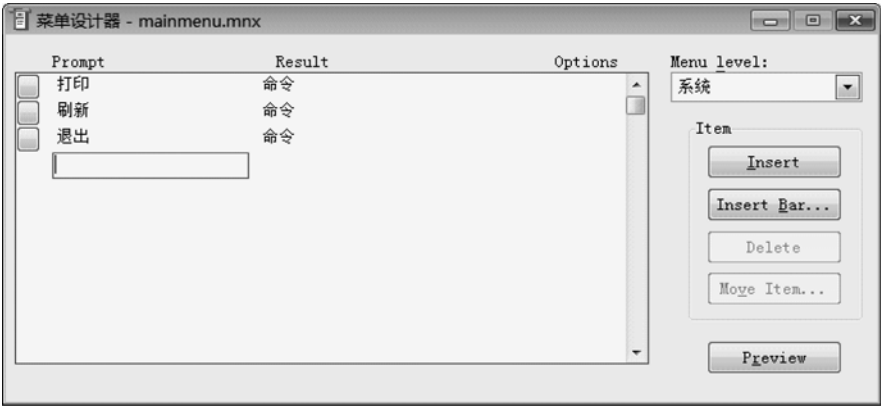


图 9-17 下级菜单设置

这里的 3 个子菜单为命令，比较简单，读者可自己添加。

## 2. 创建报表

利用刚刚学过的项目 8 的知识来创建报表 telebb，如图 9-18 所示。

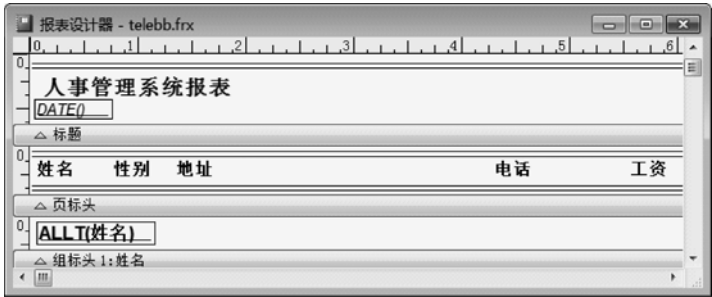


图 9-18 报表 telebb



报表预览，如图 9-19 所示。

| 姓名  | 性别 | 地址    | 电话        | 工资    | 年龄 |
|-----|----|-------|-----------|-------|----|
| 邓超  | 男  | 河北唐山市 | 6,633,537 | 4,300 | 33 |
| 李勇  | 男  | 湖南长沙市 | 2,233,530 | 3,200 | 28 |
| 卢国荣 | 男  | 江西江阴市 | 2,763,353 | 2,080 | 22 |
| 吕爱国 | 男  | 上海南汇区 | 7,343,630 | 6,800 | 35 |
| 孟芬芬 |    |       |           |       |    |

图 9-19 报表 telebb 预览

## 任务 9.3 系统完善

### 9.3.1 填写项目信息

设计好应用系统后，可以填写详细的项目信息，操作步骤如下。

(1) 打开项目文件，选择“项目”|“项目信息”命令，在“项目信息”对话框的“项目”选项卡中输入开发者的相关信息，如图 9-20 所示。

图 9-20 输入开发者的相关信息

(2) 如果在设计过程中有许多的临时表或数据表,则可以屏蔽它们,切换到“文件”选项卡,选择不用表单和数据表,如图 9-21 所示。



图 9-21 屏蔽不用表单和数据表

### 9.3.2 生成应用程序

设计完应用程序后,当然不需要他人查看代码及其设计过程,可以使系统生成可执行的程序。

操作步骤如下。

(1) 选择“项目”|“连编”命令,在“连编选项”对话框中选中“Win32 可执行程序/COM 服务程序(exe)”,“连编选项”对话框如图 9-22 所示。



图 9-22 “连编选项”对话框

(2) 单击“确定”按钮,弹出“另存为”对话框,输入可执行文件的名称“人事管理系统.exe”,如图 9-23 所示。

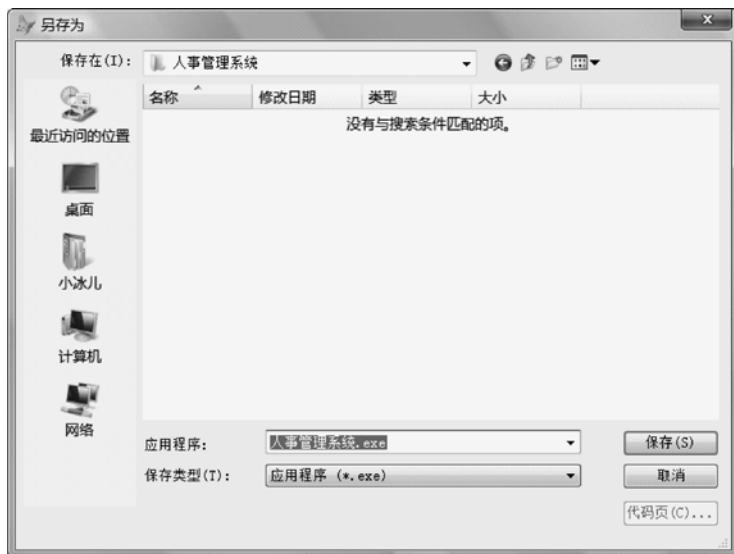


图 9-23 输入文件名“人事管理系统.exe”

(3) 单击“保存”按钮，生成应用程序“人事管理系统.exe”。

该设计过程包括了主要的操作步骤和所需要的文件，读者可根据实际需要添加文件或修改已有内容。这样，一个完整的可执行应用程序就基本设计完成了，该应用程序可用于一般单位的人事管理。

## 任务 9.4 模仿设计与自主设计

利用项目管理器组织、设计并连编一个图书管理系统应用程序，要求如下。

(1) 系统由数据库、表单、报表、菜单和程序组成。

(2) 系统中有一个数据库，数据库中至少包括 3 个数据表：图书信息表、借阅者信息表、借阅者身份表。图书信息包括图书编号、图书名称、作者、出版社、出版日期、备注、价格和数量；借阅者信息包括借书证号、姓名、性别、身份证、联系电话和密码；借阅者身份包括身份编号、身份描述和最大借阅数。

(3) 系统能够通过菜单实现数据维护、浏览、查询、报表和退出等基本功能；读者可以自行设计使系统具有更强的功能。

(4) 进入系统时应设置具备口令检查功能的主界面，该口令在进入系统后可以通过菜单中的相关命令来修改。

## 项目小结 9

项目 9 通过一个实例集中介绍了开发数据库应用程序的方法和步骤，介绍了如何把设计好的数据库、表单、报表和菜单等分离的应用系统组件，在项目管理器中连编成一个完

整的应用程序，最终编译成一个扩展名为.app 的应用文件或.exe 的可执行文件的过程，从而使读者对 Visual FoxPro 应用程序的设计方法有了一个全面的了解。

## 项目习题 9

### 1. 选择题

- (1) 连编后可以脱离 Visual FoxPro 独立运行的程序是 ( )。  
A . APP 程序    B . EXE 程序    C . FXP 程序    D . PRG 程序
- (2) 连编应用程序不能生成的文件是 ( )。  
A . APP 文件    B . EXE 文件    C . DLL 文件    D . PRG 文件
- (3) 作为整个应用程序入口点的主程序至少应具有 ( ) 功能。  
A . 初始化环境  
B . 初始化环境、显示初始用户界面  
C . 初始化环境、显示初始用户界面、控制事件循环  
D . 初始化环境、显示初始用户界面、控制事件循环、退出时恢复系统环境

### 2. 填空题

- (1) 在 Visual FoxPro 中，BUILD\_\_\_\_\_命令连编生成的程序可以脱离 Visual FoxPro 在 Windows 环境下运行。
- (2) 在面向对象方法中，类的实例称为\_\_\_\_\_。
- (3) 使用“应用程序向导”创建的项目，除项目外还自动生成\_\_\_\_\_。
- (4) 在应用程序生成器的“常规”选项卡中，选择程序类型时选中“顶层”单选按钮，将生成一个\_\_\_\_\_。
- (5) 在应用程序生成器的“常规”选项卡中，选择程序类型时选中“正常”单选按钮，将生成一个\_\_\_\_\_。
- (6) 要使得在“应用程序生成器”中所做的修改与当前活动项目保持一致，应单击\_\_\_\_\_按钮。



## 项目习题参考答案

### 项目 1 答案

#### 1. 选择题

- |        |        |        |        |        |
|--------|--------|--------|--------|--------|
| (1) B  | (2) B  | (3) D  | (4) D  | (5) C  |
| (6) C  | (7) D  | (8) B  | (9) C  | (10) A |
| (11) C | (12) D | (13) D | (14) B | (15) C |

#### 2. 填空题

- |                |                   |
|----------------|-------------------|
| (1) N 或数值型     | (2) {^1962-10-27} |
| (3) ABCD       | (4) QUIT          |
| (5) .PJX 或.pjx | (6) 移去 删除         |

### 项目 2 答案

#### 1. 选择题

- |       |       |       |       |        |
|-------|-------|-------|-------|--------|
| (1) B | (2) B | (3) D | (4) B | (5) A  |
| (6) A | (7) D | (8) B | (9) C | (10) D |

#### 2. 填空题

- |                   |                 |        |
|-------------------|-----------------|--------|
| (1) 层次模型          | (2) 事物之间的联系     | (3) 空值 |
| (4) 数据库管理系统或 DBMS | (5) 去掉重复属性的等值连接 |        |

### 项目 3 答案

#### 1. 选择题

- |        |        |        |        |        |
|--------|--------|--------|--------|--------|
| (1) C  | (2) B  | (3) B  | (4) A  | (5) D  |
| (6) D  | (7) B  | (8) D  | (9) C  | (10) D |
| (11) A | (12) A | (13) C | (14) D | (15) B |

#### 2. 填空题

- |           |                     |           |
|-----------|---------------------|-----------|
| (1) 级联    | (2) SET RELATION TO | (3) 逻辑表达式 |
| (4) 32767 | (5) REMOVE          | (6) TO A  |

### 项目 4 答案

#### 1. 选择题

- |        |        |        |        |        |
|--------|--------|--------|--------|--------|
| (1) C  | (2) D  | (3) B  | (4) B  | (5) C  |
| (6) C  | (7) D  | (8) A  | (9) B  | (10) D |
| (11) A | (12) A | (13) C | (14) B | (15) D |

#### 2. 填空题

- |           |                     |
|-----------|---------------------|
| (1) CHECK | (2) DELETE 和 UPDATE |
|-----------|---------------------|

- (3) SOME 和 WHERE      (4) COLUMN  
(5) SUM (工资)      (6) UPDATE 和 SET

#### 项目 5 答案

##### 1. 选择题

- (1) A      (2) C      (3) B      (4) A      (5) A  
(6) B      (7) C      (8) D      (9) D      (10) C

##### 2. 填空题

- (1) 计算机等级二级 Visual Foxpro      (2) 13  
(3) 静态分析      (4) MODIFY COMMAND PRG  
(5) .T.      (6) 局部变量

#### 项目 6 答案

##### 1. 选择题

- (1) C      (2) D      (3) A      (4) B      (5) D  
(6) C      (7) A      (8) C      (9) A      (10) B

##### 2. 填空题

- (1) VISUAL      (2) RECORDSOURCE      (3) CAPTION  
(4) 前      (5) 类

#### 项目 7 答案

##### 1. 选择题

- (1) A      (2) A      (3) C      (4) D      (5) A      (6) B

##### 2. 填空题

- (1) SYSMENU      (2) RIGHTCLICK  
(3) 条形菜单 一组弹出式菜单      (4) 常规选项 Show Windows Init

#### 项目 8 答案

##### 1. 选择题

- (1) D      (2) A      (3) B      (4) B      (5) D

##### 2. 填空题

- (1) 数据源 布局      (2) 图片 通用型字段  
(3) 组标头 组注脚带区      (4) 页面设置  
(5) REPORT FORM      (6) .FRX

#### 项目 9 答案

##### 1. 选择题

- (1) B      (2) D      (3) D

##### 2. 填空题

- (1) EXE      (2) 对象  
(3) 应用程序框架      (4) .EXE 可执行程序  
(5) .APP 应用程序      (6) 清理





## 2010 年 9 月计算机二级 VFP 笔试试题

### 一、选择题 (每小题 2 分, 共 70 分)

下列各题 A)、B)、C)、D) 四个选项中, 只有一个选项是正确的。请将正确选项填涂在答题卡相应位置上, 答在试卷上不得分。

- (1) 下列叙述中正确的是 ( )。
- A) 线性表的链式存储结构与顺序存储结构所需要的存储空间是相同的
  - B) 线性表的链式存储结构所需要的存储空间一般要多于顺序存储结构
  - C) 线性表的链式存储结构所需要的存储空间一般要少于顺序存储结构
  - D) 上述三种说法都不对
- (2) 下列叙述中正确的是 ( )。
- A) 在栈中, 栈中元素随栈底指针与栈顶指针的变化而动态变化
  - B) 在栈中, 栈顶指针不变, 栈中元素随栈底指针的变化而动态变化
  - C) 在栈中, 栈底指针不变, 栈中元素随栈顶指针的变化而动态变化
  - D) 上述三种说法都不对
- (3) 软件测试的目的是 ( )。
- A) 评估软件可靠性
  - B) 发现并改正程序中的错误
  - C) 改正程序中的错误
  - D) 发现程序中的错误
- (4) 下面描述中, 不属于软件危机表现的是 ( )。
- A) 软件过程不规范
  - B) 软件开发生产率低
  - C) 软件质量难以控制
  - D) 软件成本不断提高
- (5) 软件生命周期是指 ( )。
- A) 软件产品从提出、实现、使用维护到停止使用退役的过程
  - B) 软件从需求分析、设计、实现到测试完成的过程
  - C) 软件的开发过程
  - D) 软件的运行维护过程
- (6) 面向对象方法中, 继承是指 ( )。
- A) 一组对象所具有的相似性质
  - B) 一个对象具有另一个对象的性质
  - C) 各对象之间的共同性质
  - D) 类之间共享属性和操作的机制

(7) 层次型、网状型和关系型数据库划分原则是 ( )。

- A) 记录长度
- B) 文件的大小
- C) 联系的复杂程度
- D) 数据之间的联系方式

(8) 一个工作人员可以使用多台计算机, 而一台计算机可被多个人使用, 则实体工作人员与实体计算机之间的联系是 ( )。

- A) 一对一
- B) 一对多
- C) 多对多
- D) 多对一

(9) 数据库设计中反映用户对数据要求的模式是 ( )。

- A) 内模式
- B) 概念模式
- C) 外模式
- D) 设计模式

(10) 有三个关系 R、S 和 T 如下:

| R |   |   | S |   | T |   |   |   |
|---|---|---|---|---|---|---|---|---|
| A | B | C | A | D | A | B | C | D |
| a | 1 | 2 | c | 4 | c | 1 | 1 | 1 |
| b | 2 | 1 |   |   |   |   |   |   |
| c | 3 | 1 |   |   |   |   |   |   |

则由关系 R 和 S 得到关系 T 的操作是 ( )。

- A) 自然连接
- B) 交
- C) 投影
- D) 并

(11) 在 Visual FoxPro 中, 要想将日期型或日期时间型数据中的年份用 4 位数字显示, 应当使用设置命令 ( )。

- A) SET CENTURY ON
- B) SET CENTURY TO 4
- C) SET YEAR TO 4
- D) SET YAER TO yyyy

(12) 设  $A=[6*8-2]$ 、 $B=6*8-2$ ,  $C="6*8-2"$ , 属于合法表达式的是 ( )。

- A)  $A+B$
- B)  $B+C$
- C)  $A-C$
- D)  $C-B$



(13) 假设在数据库表的表设计器中, 字符型字段“性别”已被选中, 正确的有效性规则设置是 ( )。

- A) =“男”.OR.“女”
- B) 性别=“男”.OR.“女”
- C) \$“男,女”
- D) 性别“男,女”

(14) 在当前打开的表中, 显示“书名”以“计算机”打头的所有图书, 正确的命令是 ( )。

- A) list for 书名=“计算\*”
- B) list for 书名=“计算机”
- C) list for 书名=“计算%”
- D) list where 书名=“计算机”

(15) 连续执行以下命令, 最后一条命令的输出结果是 ( )。

```
SET EXACT OFF
a="北京"
b=(a="北京交通").'
?b
```

- A) 北京
- B) 北京交通
- C) .F.
- D) 出错

(16) 设 x="123", y=123, k="y", 表达式 x+&k 的值是 ( )。

- A) 123123
- B) 246
- C) 123y
- D) 数据类型不匹配

(17) 运算结果不是 2010 的表达式是 ( )。

- A) int ( 2010.9 )
- B) round ( 2010.1.0 )
- C) ceiling ( 2010.1 )
- D) floor ( 2010.9 )

(18) 在建立表间一对多的永久联系时, 主表的索引类型必须是 ( )。

- A) 主索引或候选索引
- B) 主索引、候选索引或唯一索引
- C) 主索引、候选索引、唯一索引或普通索引
- D) 可以不建立索引

(19) 在表设计器中设置的索引包含在 ( )。

- A) 独立索引文件中
- B) 唯一索引文件中

- C) 结构复合索引文件中  
D) 非结构复合索引文件中
- (20) 假设表“学生.dbf”已在某个工作区打开，且取别名为 student。选择“学生”表所在工作区为当前工作区的命令是( )。
- A) SELECT 0  
B) USE 学生  
C) SELECT 学生  
D) SELECT student
- (21) 删除视图 myview 的命令是( )。
- A) DELETE myview  
B) DELETE VIEW myview  
C) DROP VIEW myview  
D) REMOVE VIEW myview
- (22) 下面关于列表框和组合框的陈述中，正确的是( )。
- A) 列表框可以设置成多重选择，而组合框不能  
B) 组合框可以设置成多重选择，而列表框不能  
C) 列表框和组合框都可以设置成多重选择  
D) 列表框和组合框都不能设置成多重选择
- (23) 在表单设计器环境中，为表单添加一选项按钮组：“男”或“女”。默认情况下，第一个选项按钮“男”为选中状态，此时该选项按钮组的 Value 属性值为( )。
- A) 0                  B) 1                  C) “男”              D) .T.
- (24) 在 Visual Foxpro 中，属于命令按钮属性的是( )。
- A) Parent              B) This              C) ThisForm            D) Click
- (25) 在 Visual Foxpro 中，可视类库文件的扩展名是( )。
- A) .dbf                  B) .scx                  C) .vcx                  D) .dbc
- (26) 为了在报表中打印当前时间，应该在适当区域插入一个( )。
- A) 标签控件                          B) 文本框  
C) 表达式                              D) 域控件
- (27) 在菜单设计中，可以在定义菜单名称时为菜单项指定一个访问键。指定访问键为“X”的菜单项名称定义是( )。
- A) 综合查询(<X)                      B) 综合查询(>X)  
C) 综合查询(<X)                      D) 综合查询(<X)
- (28) 假设新建了一个程序文件 myProc.prg (不存在同名的.exe、.app 和.fxp 文件)，然后在命令窗口输入命令 DO myProc，执行该程序并获得正常的结果。现在用命令 ERASE myProc.prg 删除该程序文件，然后再次执行命令 DO myProc，产生的结果是( )。
- A) 出错(找不到文件)  
B) 与第一次执行的结果相同  
C) 系统打开“运行”对话框，要求指定文件  
D) 以上都不对



- (29) 以下关于视图描述错误的是 ( )。
- A) 只有在数据库中可建立视图
  - B) 视图定义保存在视图文件中
  - C) 从用户查询的角度视图和表一样
  - D) 视图物理上不包括数据
- (30) 关闭释放表单的方法是 ( )。
- A) shut
  - B) closeform
  - C) release
  - D) close
- (31) ~ (35) 题使用如下数据表。
- 学生.DBF: 学号 (C, 8), 姓名 (C, 6), 性别 (C, 2)
- 选课.DBF: 学号 (C, 8), 课程号 (C, 3), 成绩 (N, 3)
- (31) 从“选课”表中检索成绩大于等于 60 并且小于 90 的记录信息, 正确的 SQL 命令是 ( )。
- A) SELECT \* FROM 选课 WHERE 成绩 BETWEEN 60 AND 89
  - B) SELECT \* FROM 选课 WHERE 成绩 BETWEEN 60 TO 89
  - C) SELECT \* FROM 选课 WHERE 成绩 BETWEEN 60 AND 90
  - D) SELECT \* FROM 选课 WHERE 成绩 BETWEEN 60 TO 90
- (32) 检索还未确定成绩的学生选课信息, 正确的 SQL 命令是 ( )。
- A) SELECT 学生.学号, 姓名, 选课.课程号 FROM 学生 JOIN 选课  
WHERE 学生.学号=选课.学号 AND 选课.成绩 IS NULL
  - B) SELECT 学生.学号, 姓名, 选课.课程号 FROM 学生 JOIN 选课  
WHERE 学生.学号=选课.学号 AND 选课.成绩=NULL
  - C) SELECT 学生.学号, 姓名, 选课.课程号 FROM 学生 JOIN 选课  
ON 学生.学号=选课.学号 WHERE 选课.成绩 IS NULL
  - D) SELECT 学生.学号, 姓名, 选课.课程号 FROM 学生 JOIN 选课  
ON 学生.学号=选课.学号 WHERE 选课.成绩=NULL
- (33) 假设所有的选课成绩都已确定。显示“101”号课程成绩中最高的 10% 记录信息, 正确的 SQL 命令是 ( )。
- A) SELECT \* TOP 10 FROM 选课 ORDER BY 成绩 WHERE 课程号="101"
  - B) SELECT \* PERCENT 10 FROM 选课 ORDER BY 成绩 DESC  
WHERE 课程号="101"
  - C) SELECT \* TOP 10 PERCENT FROM 选课 ORDER BY 成绩  
WHERE 课程号="101"
  - D) SELECT \* TOP 10 PERCENT FROM 选课 ORDER BY 成绩 DESC  
WHERE 课程号="101"
- (34) 假设所有学生都已选课, 所有的选课成绩都已确定。检索所有选课成绩都在 90 分以上 (含) 的学生信息, 正确的 SQL 命令是 ( )。

- A) SELECT\*FROM 学生 WHERE 学号 IN ( SELECT 学号 FROM 选课 WHERE 成绩>=90 )
- B) SELECT\*FROM 学生 WHERE 学号 NOT IN ( SELECT 学号 FROM 选课 WHERE 成绩<90 )
- C) SELECT\*FROM 学生 WHERE 学号!=ANY ( SELECT 学号 FROM 选课 WHERE 成绩<90 )
- D) SELECT\* FROM 学生 WHERE 学号=ANY ( SELECT 学号 FROM 选课 WHERE 成绩>=90 )
- (35)为“选课”表增加一个“等级”字段,其类型为C、宽度为2,正确的SQL命令是( )。
- A) ALTER TABLE 选课 ADD FIELD 等级 C (2)
- B) ALTER TABLE 选课 ALTER FIELD 等级 C (2)
- C) ALTER TABLE 选课 ADD 等级 C (2)
- D) ALTER TABLE 选课 ALTER 等级 C (2)

## 二、填空题 (每空2分,共30分)

请将每个空的正确答案写在答题卡【1】~【15】序号的横线上,答在试卷上不得分。  
注意:以命令关键字填空的必须拼写完整。

(1)一个栈的初始状态为空。首先将元素5,4,3,2,1依次入栈,然后退栈一次,再将元素A,B,C,D依次入栈,之后将所有元素全部退栈,则所有元素退栈(包括中间退栈的元素)的顺序为\_\_【1】\_\_。

(2)在长度为n的线性表中,寻找最大项至少需要比较\_\_【2】\_\_次。

(3)一棵二叉树有10个度为1的结点,7个度为2的结点,则该二叉树共有\_\_【3】\_\_个结点。

(4)仅由顺序、选择(分支)和重复(循环)结构构成的程序是\_\_【4】\_\_程序。

(5)数据库设计的四个阶段是:需求分析,概念设计,逻辑设计和\_\_【5】\_\_。

(6)Visual Foxpro 索引文件不改变表中记录的\_\_【6】\_\_顺序。

(7)表达式 score<=100 AND score>=0 的数据类型是\_\_【7】\_\_。

(8) A=10

B=20

?IF (A>B,“A 大于 B”“A 不大于 B”)

执行上述程序段,显示的结果是\_\_【8】\_\_。

(9)参照完整性规则包括更新规则、删除规则和\_\_【9】\_\_规则。

(10)如果文本框中只能输入数字和正负号,需要设置文本框的\_\_【10】\_\_属性。

(11)在SQL Select 语句中使用 Group By 进行分组查询时,如果要求分组满足指定条件,则需要使用\_\_【11】\_\_子句来限定分组。

(12)预览报表 myreport 的命令是 REPORT FORM myreport\_\_【12】\_\_。

(13)将“学生”表中学号左4位为“2010”的记录存储到新表 new 中的命令是:  
SELECT\*FROM 学生 WHERE\_\_【13】\_\_="2010"\_\_【14】\_\_DBF new

(14)将“学生”表中的“学号”字段的宽度由原来的10改为12(字符型),应使用的命令是:ALTER TABLE 学生\_\_【15】\_\_。



## 2010 年 9 月计算机二级 VFP 笔试试题 参考答案

### 一、选择题

|      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| (1)  | (2)  | (3)  | (4)  | (5)  | (6)  | (7)  | (8)  | (9)  | (10) |
| B    | C    | D    | A    | A    | D    | D    | C    | C    | A    |
| (11) | (12) | (13) | (14) | (15) | (16) | (17) | (18) | (19) | (20) |
| A    | C    | D    | B    | C    | D    | C    | A    | C    | D    |
| (21) | (22) | (23) | (24) | (25) | (26) | (27) | (28) | (29) | (30) |
| C    | A    | B    | A    | C    | D    | C    | B    | B    | C    |
| (31) | (32) | (33) | (34) | (35) |      |      |      |      |      |
| A    | C    | D    | B    | C    |      |      |      |      |      |

### 二、填空题

- (1) 1DCBA2345
- (2) n-1
- (3) 25
- (4) 结构化
- (5) 物理设计
- (6) 物理
- (7) 逻辑型
- (8) A 不大于 B
- (9) 插入
- (10) InputMask
- (11) Having
- (12) Preview
- (13) Left(学号,4)或 substr(学号,1,4), into
- (14) alter 学号 C(12)

# 2011 年计算机二级考试 VFP 考试大纲

## 基本要求

- (1) 具有数据库系统的基础知识。
- (2) 基本了解面向对象的概念。
- (3) 掌握关系数据库的基本原理。
- (4) 掌握数据库程序设计方法。
- (5) 能够使用 VISUAL FOXPRO 建立一个小型数据库应用系统。

## 基础知识

### 1. 基本概念

数据库、数据模型、数据库管理系统、类和对象、事件、方法。

### 2. 关系数据库

(1) 关系数据库：关系模型、关系模式、关系、元组、属性、域、主关键字和外部关键字。

- (2) 关系运算：选择、投影、连接。
- (3) 数据的一致性和完整性：实体完整性、域完整性、参照完整性。

### 3. VISUAL FOXPRO 系统特点与工作方式

- (1) WINDOWS 版本数据库的特点。
- (2) 数据类型和主要文件类型。
- (3) 各种设计器和向导。
- (4) 工作方式：交互方式（命令方式、可视化操作）和程序运行方式。

### 4. VISUAL FOXPRO 的基本数据元素

- (1) 常量、变量、表达式。
- (2) 常用函数：字符处理函数、数值计算函数、日期时间函数、数据类型转换函数、测试函数。

### 一、VISUAL FOXPRO 数据库的基本操作

#### 1. 数据库和表的建立、修改与有效性检验

- (1) 表结构的建立与修改。
- (2) 表记录的浏览、增加、删除与修改。
- (3) 创建数据库，向数据库添加或移出表。
- (4) 设定字段级规则和记录规则。
- (5) 表的索引：主索引、候选索引、普通索引、唯一索引。

#### 2. 多表操作

- (1) 选择工作区。





(2) 建立表之间的关联：一对一的关联；一对多的关联。

(3) 设置参照完整性。

(4) 建立表间临时关联。

### 3. 建立视图与数据查询

(1) 查询文件的建立、执行与修改。

(2) 视图文件的建立、查看与修改。

(3) 建立多表查询。

(4) 建立多表视图。

## 二、关系数据库标准语言 SQL

### 1. SQL 的数据定义功能

(1) CREATE TABLE – SQL

(2) ALTER TABLE – SQL

### 2. SQL 的数据修改功能

(1) DELETE – SQL

(2) INSERT – SQL

(3) UPDATE – SQL

### 3. SQL 的数据查询功能

(1) 简单查询。

(2) 嵌套查询。

(3) 连接查询。包括内连接，外连接，左连接，右连接，完全连接。

(4) 分组与计算查询。

(5) 集合的并运算。

## 三、项目管理器、设计器和向导的使用

### 1. 使用项目管理器

(1) 使用“数据”选项卡。

(2) 使用“文档”选项卡。

### 2. 使用表单设计器

(1) 在表单中加入和修改控件对象。

(2) 设定数据环境。

### 3. 使用菜单设计器

(1) 建立主选项。

(2) 设计。

(3) 设定菜单选项程序代码。

### 4. 使用报表设计器

(1) 生成快速报表。

(2) 修改报表布局。

(3) 设计分组报表。

(4) 设计多栏报表。

5. 使用应用程序向导

6. 应用程序生成器与连编应用程序

#### 四、VISUAL FOXPRO 程序设计

1. 命令文件的建立与运行

(1) 程序文件的建立。

(2) 简单的交互式输入、输出命令。

(3) 应用程序的调试与执行。

2. 结构化程序设计

(1) 顺序结构程序设计。

(2) 选择结构程序设计。

(3) 循环结构程序设计。

3. 过程与过程调用

(1) 子程序设计及调用。

(2) 过程与过程文件。

(3) 局部变量和全局变量，过程调用中的参数传递。

4. 用户定义对话框(MESSAGEBOX)的使用

## 参 考 文 献

- [1] 李俊德. Visual FoxPro 中文版入门与实例[M]. 北京: 清华大学出版社, 2004.
- [2] 王利. 全国计算机等级考试二级教程——Visual FoxPro 程序设计[M]. 北京: 高等教育出版社, 2001.
- [3] 萨师煊, 王珊. 数据库系统概论 (第三版) [M]. 北京: 高等教育出版社, 2000.
- [4] 顾浩. Visual FoxPro 6.0 程序设计基础上机教程[M]. 北京: 中国水利水电出版社, 2000.
- [5] 高伟, 陈林等. Visual FoxPro 9.0 基础教程[M]. 北京: 清华大学出版社, 2005.
- [6] 王祥仲. Visual FoxPro 9.0 实用培训教程[M]. 北京: 清华大学出版社, 2005.
- [7] 张武军. Visual FoxPro 8.0 快速上手培训教程[M]. 西安: 西安电子科技大学出版社, 2003.
- [8] 柳青等. Visual FoxPro 程序设计教程[M]. 北京: 高等教育出版社, 2006.
- [9] 魏绍谦. Visual FoxPro 6.0 简明教程[M]. 北京: 高等教育出版社, 2007.
- [10] 谢维成. Visual FoxPro 8.0 实用教程[M]. 北京: 清华大学出版社, 2005.
- [11] 谢膺白. 数据库基础与 Visual FoxPro 9.0 程序设计[M]. 西安: 西安电子科技大学出版社, 2008.